P89LPC952 使用指南

1.概述 5
1.1 管脚配置 5
1.2 管脚描述 6
1.3 特殊功能寄存器 11
1.4 存储器结构 17
2.时钟 18
2.1 增强型CPU 18
2.2 时钟定义 18
2.2.1 振荡器时钟（OSCCLK） 18
2.2.2 低频振荡器选项 18
2.2.3 中频振荡器选项 18
2.2.4 高频振荡器选项 18
2.3 时钟输出 19
2.4 片内RC振荡器选项 19
2.5 看门狗振荡器 19
2.6 外部时钟输入选项 20
2.7 振荡器时钟唤醒延迟 20
2.8 CPU时钟（CCLK）调整：DIVM寄存器 21
2.9 低功耗选择 21
3.A/D转换器 21
3.1 概述 21
3.2 A/D特性 21
3.2.1 A/D工作模式 22
3.2.1.1 固定通道，单次转换模式 22
3.2.1.2 固定通道，连续转换模式 22
3.2.1.3 自动扫描，单次转换模式 23
3.2.1.4 自动扫描，连续转换模式 23
3.2.1.5 双通道，连续转换模式 23
3.2.1.6 单步模式 24
3.2.2 转换模式选择位 24
3.2.3 转换启动模式 24
3.2.3.1 定时器触发启动 24
3.2.3.2 立即启动 24
3.2.3.3 边沿触发 24
3.2.4 停止和重启转换 25
3.2.5 边界限制中断 25
3.2.6 时钟分频器 25
3.2.7 用作ADC功能的I/O脚 25

3.2.8 掉电和空闲模式.....25
4 中断.....28
4.1 中断优先级结构.....29
4.2 外部中断管脚干扰抑制.....29
5.I/O口.....31
5.1 端口配置.....31
5.2 准双向口输出配置.....31
5.3 开漏输出配置.....32
5.4 仅为输入配置.....33
5.5 推挽输出配置.....33
5.6 P0 口和模拟比较器功能.....33
5.7 附加端口特性.....34
6.电源监控功能.....35
6.1 掉电检测.....35
6.2 上电检测.....36
6.3 节电模式.....36
7.复位.....39
7.1 复位向量.....41
8.定时器／计数器 0 和 1.....41
8.1 模式 0.....43
8.2 模式 1.....43
8.3 模式 2.....43
8.4 模式 3.....43
8.5 模式 6.....43
8.6 定时器溢出触发输出.....45
9.实时时钟系统定时器.....46
9.1 实时时钟源.....46
9.2 RTCS1/RTCS0 的更改.....46
9.3 实时时钟中断/唤醒.....47
9.4 影响实时时钟的复位源.....47
10.URAT.....49
10.1 模式 0.....49
10.2 模式 1.....49
10.3 模式 2.....49
10.4 模式 3.....49
10.5 SFR空间.....50
10.6 波特率发生器及选择.....50
10.7 更新BRGR1 和BRGR0 SFR寄存器.....50
10.8 帧错误.....52
10.9 间隔检测.....52
10.10 更多关于UART模式 0 的信息.....56
10.11 更多关于UART模式 1 的信息.....56
10.12 更多关于模式 2 和模式 3 的信息.....57
10.13 模式 2 和 3 中SM2_n=1 时的帧错误和RI_n.....57

10.14 间隔检测.....57
10.15 双缓冲.....58
10.16 不同模式中的双缓冲.....58
10.17 双缓冲使能时的发送中断（模式 1，2 和 3）.....58
10.18 双缓冲中的第 9 位数据（模式 1、2 和 3）.....59
10.19 多机通信.....60
10.20 自动地址识别.....60
11.I²C接口.....61
11.1 I²C总线数据寄存器.....62
11.2 I²C从地址寄存器.....62
11.3 I²C总线控制寄存器.....63
11.4 I²C状态寄存器.....64
11.5 I²C SCL占空比寄存器I2SCLH和I2SCLL.....65
11.6 I²C操作模式.....66
11.6.1 主发送器模式.....66
11.6.2 主接收器模式.....66
11.6.3 从接收器模式.....67
11.6.4 从发送器模式.....68
12.串行外围接口（SPI）.....73
12.1 配置SPI.....77
12.2 作为从机时的额外注意事项.....77
12.3 作为主机时的额外注意事项.....77
12.4 通过$\overline{SS}$改变模式.....78
12.5 写冲突.....78
12.6 数据模式.....78
12.7 SPI时钟预分频器选择.....80
13.模拟比较器.....81
13.1 比较器配置.....81
13.2 内部参考电压.....82
13.3 比较器输入管脚.....82
13.4 比较器中断.....82
13.5 比较器和节电模式.....82
13.6 比较器配置举例.....83
14.键盘中断（KBI）.....84
15.看门狗定时器(WDT).....85
15.1 看门狗功能.....85
15.2 看门狗清零序列.....86
15.3 看门狗时钟源.....89
15.4 定时器模式中的看门狗定时器.....89
15.5 掉电操作.....90
15.6 无外部振荡器的掉电周期唤醒.....90
16.附加特性.....90
16.1 软件复位.....91
16.2 双数据指针.....91

17.Flash程序存储器 92
17.1 概述 92
17.2 特性 92
17.3 Flash编程和擦除 92
17.4 Flash用作数据存储器：IAP-Lite 93
17.5 在电路编程（ICP） 96
17.6 P89LPC952 的ISP和IAP性能 96
17.7 引导ROM 97
17.8 上电复位代码的执行 97
17.9 引导装载程序的硬件激活 97
17.10 在系统编程(ISP) 98
17.11 使用在系统编程(ISP) 98
17.12 在应用中编程的方法(IAP) 102
17.13 IAP特许关键值 102
17.14 Flash写使能 102
17.15 配置字节保护 103
17.16 IAP错误状态 103
17.17 用户配置字节 106
17.18 用户保密字节 107
17.19 引导向量寄存器 108
17.20 引导状态寄存器 108
18.调试接口 109
19.指令集 110
20.修订记录 113

1.概述

P89LPC952 是一款单片封装的微控制器，适合于许多要求高集成、低成本的场合，可以满足多方面的性能要求。P89LPC952 采用高性能的微处理器结构，指令执行时间只需 2～4 个时钟周期，6 倍于标准 80C51 器件。P89LPC952 集成了许多系统级的功能，这样可大大减少元件数目、电路板面积以及系统的成本。

1.1 管脚配置

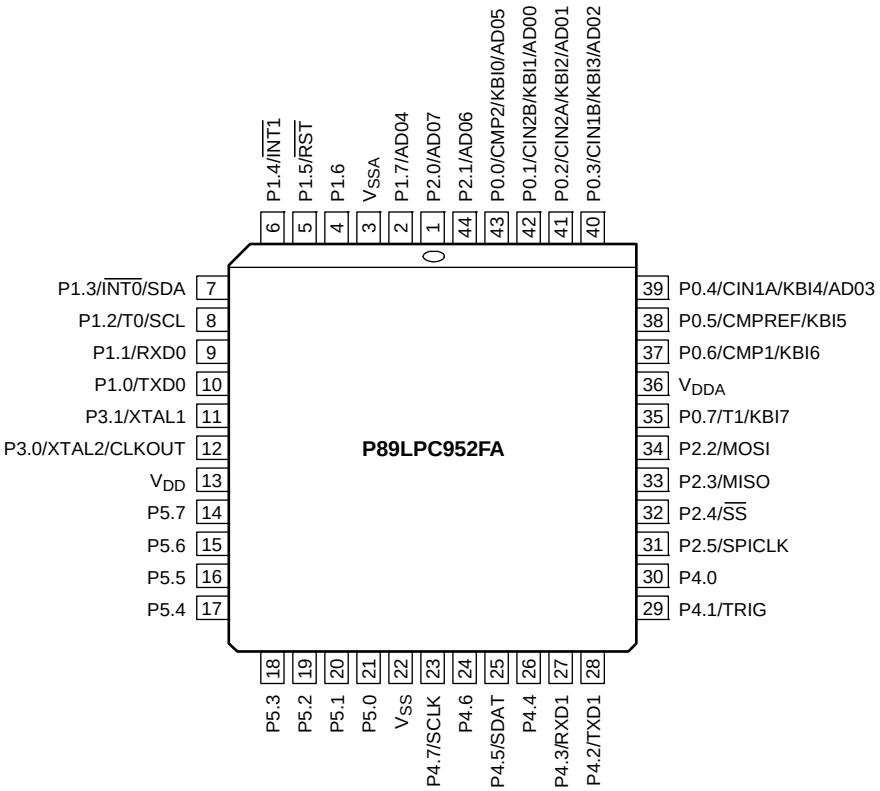


图 1 PLCC44 管脚配置

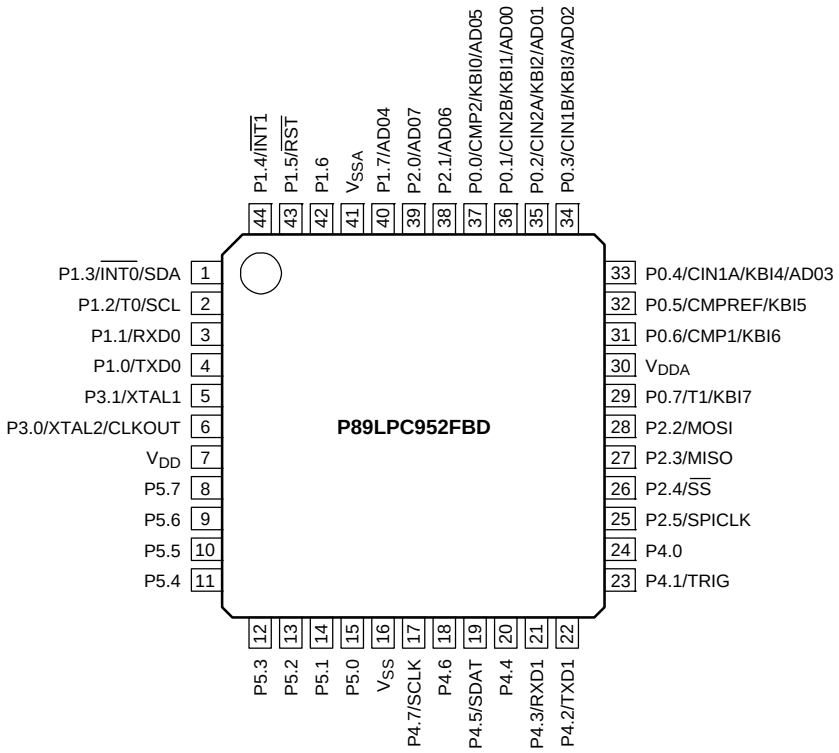


图 2 LQFP44 管脚配置

1.2 管脚描述

表 1 管脚描述

符号	管脚		类型	描述
P0.0~P0.1			I/O	**PROT0：**P0 是一个可由用户定义输出类型的 8 位 I/O 口，在上电复位时，P0 锁存器配置为内部上拉禁止的仅为输入模式。P0 口由口配置寄存器设定为输出或输入模式，每一个管脚均可单独设定。详细请参考 5.1 节“端口配置”部分。 P0 口具有键盘输入中断功能。 所有管脚都具有施密特触发输入。 P0 口还可提供如下特殊功能：
P0.0/CMP2/ KBI0/AD05	43	37	I/O O I I	**P0.0** P0 口位 0 **CMP2** 比较器 2 输出 **KBI0** 键盘输入 0 **AD05** ADC0 通道 5 模拟输入
P0.1/CIN2B/ KBI1/AD00	42	36	I/O I I I	**P0.1** P0 口位 1 **CIN2B** 比较器 2 正向输入 B **KBI1** 键盘输入 1 **AD00** ADC0 通道 0 模拟输入
P0.2/CIN2A/ KBI2/AD01	41	35	I/O I I I	**P0.2** P0 口位 2 **CIN2A** 比较器 2 正向输入 A **KBI2** 键盘输入 2 **AD01** ADC0 通道 1 模拟输入

续表 1

符号	管脚		类型	描述
P0.3/CIN1B/ KBI3/AD02	40	34	I/O I I I	**P0.3** P0 口位 3 **CIN1B** 比较器 1 正向输入 B **KBI3** 键盘输入 3 **AD02** ADC0 通道 2 模拟输入
P0.4/CIN1A/ KBI4/AD03	39	33	I/O I I I	**P0.4** P0 口位 4 **CIN1A** 比较器 1 正向输入 A **KBI4** 键盘输入 4 **AD03** ADC0 通道 3 模拟输入
P0.5/ CMPREF/ KBI5	38	32	I/O I I	**P0.5** P0 口位 5 **CMPREF** 比较器参考（反向）输入 **KBI5** 键盘输入 5
P0.6/CMP1/ KBI6	37	31	I/O I I	**P0.6** P0 口位 6 **CMP1** 比较器 1 输出 **KBI6** 键盘输入 6
P0.7/T1/ KBI7	35	29	I/O I/O I	**P0.7** P0 口位 7 **T1** 定时/计数器 1 外部计数输入或溢出输出 **KBI7** 键盘输入 7
P1.0~P1.7			I/O,I	**PORT1**：除了下面说明的三个管脚外，P1 是一个可由用户定义输出类型的 8 位 I/O 口。在上电复位时，P1 锁存器配置为内部上拉禁止的仅为输入模式。P1 口由口配置寄存器设定为输出或输入模式，每一位均可单独设定。详细请参考 5.1 节“端口配置”部分。 P1.2-P1.3 作为输出时为开漏。P1.5 为仅为输入模式。 所有管脚都具有施密特触发输入。 P1 口还可提供如下特殊功能：
P1.0/TxD0	10	4	I/O O	**P1.0** P1 口位 0 **TxD0** 串行口 0 发送器输出
P1.1/RxD0	9	3	I/O I	**P1.1** P1 口位 1 **RxD0** 串行口 0 接收器输入
P1.2/T0/SCL	8	2	I/O I/O I/O	**P1.2** P1 口位 2（作为输出时为开漏） **T0** 定时/计数器 0 外部计数输入或溢出输出(作为输出时为开漏) **SCL** I^2C 总线串行时钟输入/输出
P1.3/ **$\overline{\text{INT0}}$** / SDA	7	1	I/O I I/O	**P1.3** P1 口位 3（作为输出时为开漏） **$\overline{\text{INT0}}$** 外部中断 0 输入 **SDA** I^2C 总线串行数据输入/输出
P1.4/ **$\overline{\text{INT1}}$**	6	44	I I	**P1.4** P1 口位 4 **$\overline{\text{INT1}}$** 外部中断 1 输入。

续表 1

符号	管脚		类型	描述
P1.5/ $\overline{\textbf{RST}}$	5	43	I I	**P1.5** P1 口位 5（仅为输入） $\overline{\textbf{RST}}$ 上电或通过 UCFG1 选择作为外部复位输入。作为复位管脚时，输入的低电平会使芯片复位, I/O 口和外围功能进入默认状态，处理器从地址 0 开始执行。另外该管脚还可用于在上电时强制进入 ISP 模式。**当使用频率高于 12MHz 的振荡器时，必须使能 P1.5 的复位输入功能。上电时，需要一个外部电路使器件保持复位状态，直至 V_{DD} 到达指定的电平。当系统电源被移走时，V_{DD} 将降至低于指定的最低工作电压。在某些使用频率高于 12MHz 的振荡器的应用中，当 V_{DD} 降至低于指定的最低工作电压时，需要一个外部掉电检测电路使器件保持复位状态。**
P1.6	4	42	I/O	**P1.6** P1 口位 6
P1.7/AD04	2	40	I/O I	**P1.7** P1 口位 7 **AD04** ADC0 通道 4 模拟输入
P2.0~P2.5			I/O	**PORT2**：P2 是一个可由用户定义输出类型的 8 位 I/O 口，在上电复位时，P2 锁存器配置为内部上拉禁止的仅为输入模式。P2 口由口配置寄存器设定为输出或输入模式，每一位均可单独设定。详细请参考 5.1 节“端口配置”部分。 所有管脚都具有施密特触发输入。 P2 口还可提供如下特殊功能：
P2.0/AD07	1	39	I/O I	**P2.0** P2 口位 0 **AD07** ADC0 通道 7 模拟输入
P2.1/AD06	44	38	I/O I	**P2.1** P2 口位 1 **AD06** ADC0 通道 6 模拟输入
P2.2/MOSI	34	28	I/O I/O	**P2.2** P2 口位 2 **MOSI** SPI 主机输出/从机输入。当配置为主机时，该管脚为输出；当配置为从机时，该管脚为输入。
P2.3/MISO	33	27	I/O I/O	**P2.3** P2 口位 3 **MISO** 当配置为主机时，该管脚为输入；当配置为从机时，该管脚为输出。
P2.4/ $\overline{\textbf{SS}}$	32	26	I/O I/O	**P2.4** P2 口位 4 $\overline{\textbf{SS}}$ SPI 从机选择
P2.5/ SPICLK	31	25	I/O I/O	**P2.5** P2 口位 5 **SPICLK** SPI 时钟。当配置为主机时，该管脚为输出；当配置为从机时，该管脚为输入
P3.0~P3.1			I/O	**PORT3**：P3 是一个可由用户定义输出类型的 2 位 I/O 口，在上电复位时，P3 锁存器配置为内部上拉禁止的仅为输入模式。P3 口由口配置寄存器设定为输出或输入模式，每一个管脚均可单独设定。详细请参考 5.1 节“端口配置”部分。 所有管脚都具有施密特触发输入。 P3 口还可提供如下特殊功能：

续表 1

符号	管脚		类型	描述
P3.0/XTAL2 /CLKOUT	12	6	I/O O	**P3.0** P3 口位 3 **XTAL2** 振荡放大器输出（通过 Flash 配置为晶体振荡器时） **CLKOUT** 当该位通过 SFR（ENCLK-TRIM.6）使能时，为 CPU 时钟 2 分频后输出。当 CPU 时钟为内部 RC 振荡器、看门狗振荡器或外部时钟输入时可用，但当 XTAL1/XTAL2 用于产生 RTC/系统定时器的时钟源时除外。
P4.0~p4.7			I/O	**PROT4** P4 是一个可由用户配置输出类型的 8 位 I/O 口。在上电复位时，P4 锁存器配置为内部上拉禁止的仅为输入模式。P4 口由口配置寄存器设定为输出或输入模式，每一个管脚均可单独设定。详细请参考 5.1 节“端口配置”部分。 所有管脚都具有施密特触发输入。 P4 口还可提供如下特殊功能：
P4.0	30	24	I/O	**P4.0** P4 口位 0
P4.1/TRIG	29	23	I/O O	**P4.1** P4 口位 1 **TRIG** JTAG 触发输出
P4.2/TxD1	28	22	I/O O	**P4.2** P4 口位 2 **TxD1** 串行口 1 发送器输出
P4.3/RxD1	27	21	I/O I	**P4.3** P4 口位 3 **RxD1** 串行口 1 接收器输入
P4.4	26	20	I/O	**P4.4** P4 口位 4
P4.5/SDAT	25	19	I/O I/O	**P4.5** P4 口位 5 **SDAT** JTAG 接口的串行数据输入/输出
P4.6	24	18	I/O	**P4.6** P4 口位 6
P4.7/SCLK	23	17	I/O I	**P4.7** P4 口位 7 **SCLK** JTAG 接口的串行时钟输入
P5.0~P5.7			I/O	**PORT5** P5 是可由用户配置输出类型的 8 位 I/O 口。上电复位时，P5 锁存器配置为内部上拉禁止的仅为输入模式。P5 口由口配置寄存器设定为输入或输出模式。每个管脚可单独设定。详情请参考 5.1 节“端口配置”部分。 所有管脚都具有施密特触发输入。 P5 还提供如下特殊功能：
P5.0	21	15	I/O	**P5.0** P5 口位 0
P5.1	20	14	I/O	**P5.1** P5 口位 1
P5.2	19	13	I/O	**P5.2** P5 口位 2
P5.3	18	12	I/O	**P5.3** P5 口位 3
P5.4	17	11	I/O	**P5.4** P5 口位 4
P5.5	16	10	I/O	**P5.5** P5 口位 5
P5.6	15	9	I/O	**P5.6** P5 口位 6
P5.7	14	8	I/O	**P5.7** P5 口位 7
V_{SS}	22	16	I	**地** 0V 参考
V_{SSA}	3	41	I	**模拟地** 0V 模拟参考

续表 1

符号	管脚		类型	描述
V_{DD}	13	7	I	**电源** 这是正常工作、空闲和掉电模式下的电源电压
V_{DDA}	36	30	I	**模拟电源** 这是正常工作、空闲和掉电模式下的模拟电源电压

[1] P1.0~P1.4,P1.6,P1.7 为输入/输出，P1.5 为输入

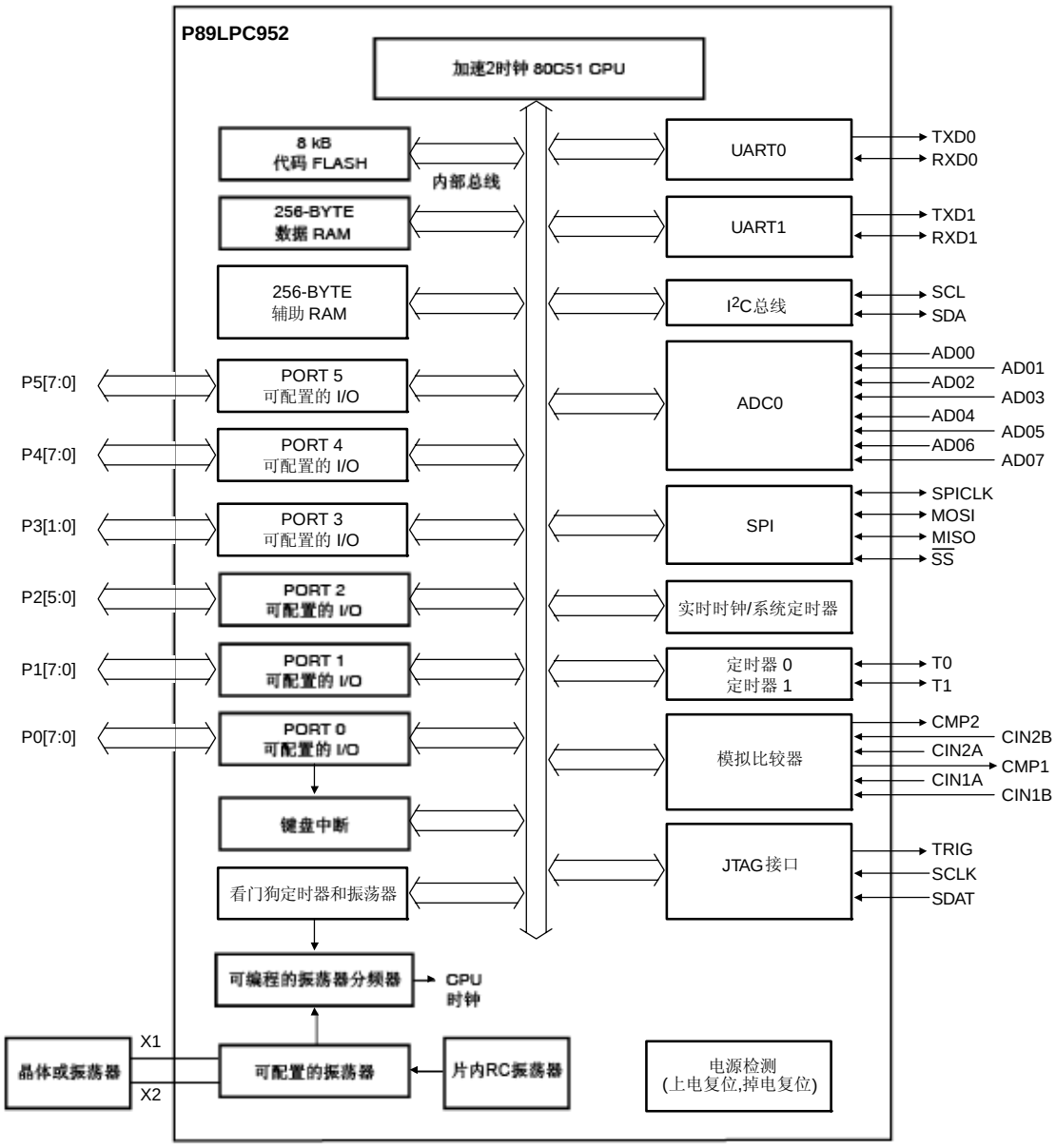


图 3 框图

1.3 特殊功能寄存器

注：对特殊功能寄存器(SFR)的访问必须遵循以下方式：

- 用户不要试图访问任何未经定义的 SFR 地址。
- 对任何已定义的 SFR 的访问必须符合 SFR 的功能。
- 标注为‘-’,‘0’或‘1’的 SFR 位只能以如下方式读或写：

  —‘-’必须写入 0，除非另有说明，但当读出时可返回任意值（即使向其写入 0）。这是一个保留位，作为将来功能扩展之用。

  —‘0’必须写入 0，并且当读出时返回 0。

  —‘1’必须写入 1，并且当读出时返回 1。

表 2 特殊功能寄存器

*表示可直接位寻址的 SFR

名称	定义	地址	位功能和位地址 MSB							LSB	复位值 Hex	Binary
		位地址	E7	E6	E5	E4	E3	E2	E1	E0		
ACC*	累加器	E0H									00	00000000
AD0CON	ADC0 控制寄存器	97H	ENBI0	ENADCI0	TMM0	EDGE0	ADCI0	ENADC0	ADCS01	ADCS00	00	00000000
AD0INS	ADC0 输入选择	A3H	ADI07	ADI06	ADI05	ADI04	ADI03	ADI02	ADI01	ADI00	00	00000000
AD0MODA	ADC0 模式寄存器 A	C0H	BNDI0	BURST0	SCC0	SCAN0	-	-	-	-	00	00000000
AD0MODB	ADC0 模式寄存器 B	A1H	CLK2	CLK1	CLK0	-	-	-	-	-	00	000x0000
AUXR1	辅助功能寄存器	A2H	CLKLP	EBRR	ENT1	ENT0	SRST	0	-	DPS	00	00000000
		位地址	F7	F6	F5	F4	F3	F2	F1	F0		
B*	B 寄存器	F0H									00	00000000
BRGR0_0	波特率发生器 0 低字节	BEH									00	00000000
BRGR1_0	波特率发生器 0 高字节	BFH									00	00000000
BRGCON_0	波特率发生器 0 控制	BDH	-	-	-	-	-	-	SBRGS_0	BRGEN_0	00[2]	xxxxxx00
CMP1	比较器 1 控制寄存器	ACH	-	-	CE1	CP1	CN1	OE1	CO1	CMF1	00[1]	xx000000
CMP2	比较器 2 控制寄存器	ADH	-	-	CE2	CP2	CN2	OE2	CO2	CMF2	00[1]	xx000000
DIVM	CPU 时钟分频控制	95H									00	00000000

续表 2

名称	定义	地址	位功能和位地址 MSB							LSB	复位值 Hex	Binary
DPTR	数据指针(双字节)											
DPH	指针高字节	83H									00	00000000
DPL	指针低字节	82H									00	00000000
FMADRH	编程 Flash 地址高字节	E7H									00	00000000
FMADRL	编程 Flash 地址低字节	E6H									00	00000000
FMCON	编程 Flash 控制（读）	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70	01110000
	编程 Flash 控制（写）	E4H	FMCMD.7	FMCMD.6	FMCMD.5	FMCMD.4	FMCMD.3	FMCMD.2	FMCMD.1	FMCMD.0		
FMDATA	编程 Flash 数据	E5H									00	00000000
I2ADR	I^2C 从地址寄存器	DBH	I2ADR.6	I2ADR.5	I2ADR.4	I2ADR.3	I2ADR.2	I2ADR.1	I2ADR.0	GC	00	00000000
		位地址	DF	DE	DD	DC	DB	DA	D9	D8		
I2CON*	I^2C 控制寄存器	D8H	-	I2EN	STA	STO	SI	AA	-	CRSEL	00	x00000x0
I2DAT	I^2C 数据寄存器	DAH										
I2SCLH	串行时钟发生器/SCL 占空比寄存器高字节	DDH									00	00000000
I2SCLL	串行时钟发生器/SCL 占空比寄存器低字节	DCH									00	00000000
I2STAT	I^2C 状态寄存器	D9H	STA.4	STA.3	STA.2	STA.1	STA.0	0	0	0	F8	11111000
		位地址	AF	AE	AD	AC	AB	AA	A9	A8		
IEN0*	中断使能 0	A8H	EA	EWDRT	EBO	ES/ESR	ET1	EX1	ET0	EX0	00	00000000
		位地址	EF	EE	ED	EC	EB	EA	E9	E8		
IEN1*	中断使能 1	E8H	-	EST	-	-	ESPI	EC	EKBI	EI2C	00[1]	00x00000
IEN2	中断使能 2	D5H	-	-	-	-	EST1	ES1/ESR1	EADC	-	00[1]	00x00000
		位地址	BF	BE	BD	BC	BB	BA	B9	B8		
IP0*	中断优先级 0	B8H	-	PWDRT	PBO	PS/PSR	PT1	PX1	PT0	PX0	00[1]	x0000000
IP0H	中断优先级 0 高字节	B7H	-	PWDRTH	PBOH	PSH/PSRH	PT1H	PX1H	PT0H	PX0H	00[1]	x0000000
		位地址	FF	FE	FD	FC	FB	FA	F9	F8		
IP1*	中断优先级 1	F8H	-	PST	-	-	PSPI	PC	PKBI	PI2C	00[1]	00x00000
IP1H	中断优先级 1 高字节	F7H	-	PSTH	-	-	PSPIH	PCH	PKBIH	PI2CH	00[1]	00x00000
IP2	中断优先级 2	D6H	-	-	-	-	PEST1	PES1/PESR1	PADC	-	00[1]	00x00000

续表 2

名称	定义	地址	位功能和位地址 MSB							LSB	复位值 Hex	Binary
IP2H	中断优先级 2 高字节	D7H	-	-	-	-	PEST1H	PES1H/PESR1H	PADCH	-	00[1]	00x00000
KBCON	键盘控制寄存器	94H	-	-	-	-	-	-	PATN_SEL	KBIF	00[1]	xxxxxx00
KBMASK	键盘中断屏蔽寄存器	86H									00	00000000
KBPATN	键盘模式	93H									FF	11111111
		位地址	87	86	85	84	83	82	81	80		
P0*	P0 口	80H	T1/KB7	CMP1/KB6	CMPREF/KB5	CIN1A/KB4	CIN1B/KB3	CIN2A/KB2	CIN2B/KB1	CMP2/KB0	[1]	
		位地址	97	96	95	94	93	92	91	90		
P1*	P1 口	90H	-	-	$\overline{RST}$	$\overline{INT1}$	$\overline{INT0}$/SDA	T0/SCL	RxD0	TxD0	[1]	
		位地址	97	96	95	94	93	92	91	90		
P2*	P2 口	A0H	-	-	SPICLK	$\overline{SS}$	MISO	MOSI	-	-	[1]	
		位地址	B7	B6	B5	B4	B3	B2	B1	B0		
P3*	P3 口	B0H	-	-	-	-	-	-	XTAL1	XTAL2	[1]	
P4	P4 口	B3	-	TMS	-	-	RxD1	TxD1	TRIG	T3EX	[1]	
P5	P5 口	B4		T3	-	-	-	-	-	-	[1]	
P0M1	P0 口输出模式 1	84H	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	FF[1]	11111111
P0M2	P0 口输出模式 2	85H	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0	00[1]	00000000
P1M1	P1 口输出模式 1	91H	P1M1.7	P1M1.6	-	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	D3[1]	11x1xx11
P1M2	P1 口输出模式 2	92H	P1M2.7	P1M2.6	-	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	00[1]	00x0xx00
P2M1	P2 口输出模式 1	A4H	-	-	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	FF[1]	11111111
P2M2	P2 口输出模式 2	A5H	-	-	P2M2.5	P2M2.4	P2M2.3	P2M2.2	P2M2.1	P2M2.0	00[1]	00000000
P3M1	P3 口输出模式 1	B1H	-	-	-	-	-	-	P3M1.1	P3M1.0	03[1]	xxxxxx11
P3M2	P3 口输出模式 2	B2H	-	-	-	-	-	-	P3M2.1	P3M2.0	00[1]	xxxxxx00
PCON	电源控制寄存器	87H	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00	00000000
PCONA	电源控制寄存器 A	B5H	RTCPD	DEEPD	VCPD	ADPD	I2PD	SPPD	SPD	CCUPD	00[1]	00000000
		位地址	D7	D6	D5	D4	D3	D2	D1	D0		
PSW*	程序状态字	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00	00000000

续表 2

名称	定义	地址	位功能和位地址 MSB							LSB	复位值 Hex	Binary
PT0AD	0 口数字输入禁能	F6H	-	-	PT0AD.5	PT0AD.4	PT0AD.3	PT0AD.2	PT0AD.1	-	00	xx00000x
RSTSRC	复位源寄存器	DFH	-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX	[3]	
RTCCON	RTC 控制	D1H	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60[1][6]	011xxx00
RTCH	RTC 寄存器高字节	D2H									00[6]	00000000
RTCL	RTC 寄存器低字节	D3H									00[6]	00000000
S0ADDR	串口地址寄存器	A9H									00	00000000
S0ADEN	串口地址使能	B9H									00	00000000
S0BUF	串口数据缓冲区寄存器	99H									xx	xxxxxxxx
		位地址	9F	9E	9D	9C	9B	9A	99	98		
S0CON*	串行口控制	98H	SM0_0/FE_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0	00	00000000
S0STAT	串行口扩展状态寄存器	BAH	DBMOD_0	INTLO_0	CIDIS_0	DBISEL_0	FE_0	BR_0	OE_0	STINT_0	00	00000000
SP	堆栈指针	81H									07	00000111
SPCTL	SPI 控制寄存器	E2H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	04	00000100
SPSTAT	SPI 状态寄存器	E1H	SPIF	WCOL	-	-	-	-	-	-	00	00xxxxxx
SPDAT	SPI 数据寄存器	E3H									00	00000000
S1CON	串行口 1 控制	B6H	SM0_1/FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	00	00000000
S1STAT	串行口 1 扩展状态寄存器	D4H	DBMOD_1	INTLO_1	CIDIS_1	DBISEL_1	FE_1	BR_1	OE_1	STINT_1	00	00000000
TAMOD	定时器 0/1 辅助模式	8FH	-	-	-	T1M2	-	-	-	T0M2	00	xxx0xxx0
		位地址	8F	8E	8D	8C	8B	8A	89	88		
TCON*	定时器 0/1 控制	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00	00000000
TH0	定时器 0 高字节	8CH									00	00000000
TH1	定时器 1 高字节	8DH									00	00000000
TL0	定时器 0 低字节	8AH									00	00000000

续表 2

名称	定义	地址	位功能和位地址 MSB							LSB	复位值 Hex	Binary
TL1	定时器 1 低字节	8BH									00	00000000
TL2	CCU 定时器低字节	CCH									00	00000000
TMOD	定时器 0/1 工作模式	89H	T1GATE	TIC/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00	00000000
TRIM	内部振荡调整寄存器	96H	RCCLK	ENCLK	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0	[5] [6]	
WDCON	看门狗控制寄存器	A7H	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK	[4] [6]	
WDL	看门狗装载	C1H									FF	11111111
WFEED1	看门狗清零 1	C2H										
WFEED2	看门狗清零 2	C3H										

注：

[1] 上电复位后所有的 I/O 口都处于仅为输入(高阻)状态。

[2] 只有当 BRGCON_0 SFR 中的 BRGEN_0 为 0 时，才可对 BRGR1_0 和 BRGR0_0 进行写操作。当 BRGEN_0=1 时，写入其中任意一个，结果将是不可预知的。

[3] RSTSRC 寄存器反映 UM1.147 复位的原因。在上电复位时，所有复位源标志都清零(POF 和 BOF 除外)。上电复位值为 xx110000。

[4] 复位后 WDCON 的值为 111001x1，即 PRE2-PRE0 全 1，WDRUN=1，WDCLK=1。看门狗复位后，WDTOF=1; 上电复位后，WDTOF=0。其它复位不会影响 WDTOF。

[5] 上电复位时，TRIM 寄存器初始化为出厂时的配置。其它复位不会使 TRIM 寄存器初始化。

[6] 上电复位是唯一影响这些特殊功能寄存器的复位。

表 3 扩展的特殊功能寄存器

名称	描述	SFR 地址	位功能和地址 MSB	LSB	复位值 Hex	Binary
ADC0HBND	ADC0 高_边界寄存器，左（MSB）	FFEFh			FF	1111 1111
ADC0LBND	ADC0 低_边界寄存器（MSB）	FFEEh			00	0000 0000
AD0DAT0R	ADC0 数据寄存器 0，右（LSB）	FFFEh	AD0DAT0[7:0]		00	0000 0000
AD0DAT0L	ADC0 数据寄存器 0，左（MSB）	FFFFh	AD0DAT0[9:2]		00	0000 0000
AD0DAT1R	ADC0 数据寄存器 1，右（LSB）	FFFCh	AD0DAT1[7:0]		00	0000 0000
AD0DAT1L	ADC0 数据寄存器 1，左（MSB）	FFFDh	AD0DAT1[9:2]		00	0000 0000

续表 3

名称	描述	SFR 地址	位功能和地址 MSB LSB	复位值 Hex Binary
AD0DAT2R	ADC0 数据寄存器 2，右（LSB）	FFFAh	AD0DAT2[7:0]	00 0000 0000
AD0DAT2L	ADC0 数据寄存器 2，左（MSB）	FFFBh	AD0DAT2[9:2]	00 0000 0000
AD0DAT3R	ADC0 数据寄存器 3，右（LSB）	FFF8h	AD0DAT3[7:0]	00 0000 0000
AD0DAT3L	ADC0 数据寄存器 3，左（MSB）	FFF9h	AD0DAT3[9:2]	00 0000 0000
AD0DAT4R	ADC0 数据寄存器 4，右（LSB）	FFF6h	AD0DAT4[7:0]	00 0000 0000
AD0DAT4L	ADC0 数据寄存器 4，左（MSB）	FFF7h	AD0DAT4[9:2]	00 0000 0000
AD0DAT5R	ADC0 数据寄存器 5，右（LSB）	FFF4h	AD0DAT5[7:0]	00 0000 0000
AD0DAT5L	ADC0 数据寄存器 5，左（MSB）	FFF5h	AD0DAT5[9:2]	00 0000 0000
AD0DAT6R	ADC0 数据寄存器 6，右（LSB）	FFF2h	AD0DAT6[7:0]	00 0000 0000
AD0DAT6L	ADC0 数据寄存器 6，左（MSB）	FFF3h	AD0DAT6[9:2]	00 0000 0000
AD0DAT7R	ADC0 数据寄存器 7，右（LSB）	FFF0h	AD0DAT7[7:0]	
AD0DAT7L	ADC0 数据寄存器 7，左（MSB）	FFF1h	AD0DAT7[9:2]	
BNDSTA0	ADC0 边界状态寄存器	FFEDh		
BRGCON_1	波特率发生器 1 控制	FFB3h	- - - - - - SBRGS_1 BRGEN_1	00[2]xxxx xx00
BRG0_1	波特率发生器 1 低字节	FFB4h		
BRG1_1	波特率发生器 1 高字节	FFB5h		
P4M1	P4 输出模式 1	FFB8h	(P4M1.7) (P4M1.6) (P4M1.5) (P4M1.4) (P4M1.3) (P4M1.2) (P4M1.1) (P4M1.0)	FF[1]1111 1111
P4M2	P4 输出模式 2	FFB9h	(P4M2.7) (P4M2.6) (P4M2.5) (P4M2.4) (P4M2.3) (P4M2.2) (P4M2.1) (P4M2.0)	00[1]0000 0000
P5M2	P5 输出模式 1	FFBAh	(P5M1.7) (P5M1.6) (P5M1.5) (P5M1.4) (P5M1.3) (P5M1.2) (P5M1.1) (P5M1.0)	FF[1]1111 1111
P5M2	P5 输出模式 2	FFBBh	(P5M2.7) (P5M2.6) (P5M2.5) (P5M2.4) (P5M2.3) (P5M2.2) (P5M2.1) (P5M2.0)	00[1]0000 0000
S1ADDR	串行口 1 地址寄存器	FFB2h		00 0000 0000
S1ADEN	串行口 1 地址使能	FFB1h		00 0000 0000
S1BUF	串行口 1 数据缓冲器寄存器	FFB0h		xx xxxx xxxx

[1] 扩展的 SFR 在物理上位于片内，但逻辑上位于外部数据存储器地址空间（XDATA）。对这些扩展的 SFR 的寻址使用 MOVX A ,@DPTR 和 MOVX @DPTR,A 指令。

[2] 只有当 BRGCON_1 SFR 的 BRGEN_1 为 0 时，BRGR1_1 和 BRGR0_1 才能执行写入。当 BRGEN_1=1 时写入其中任何一个，结果将是不可预知的。

1.4 存储器结构

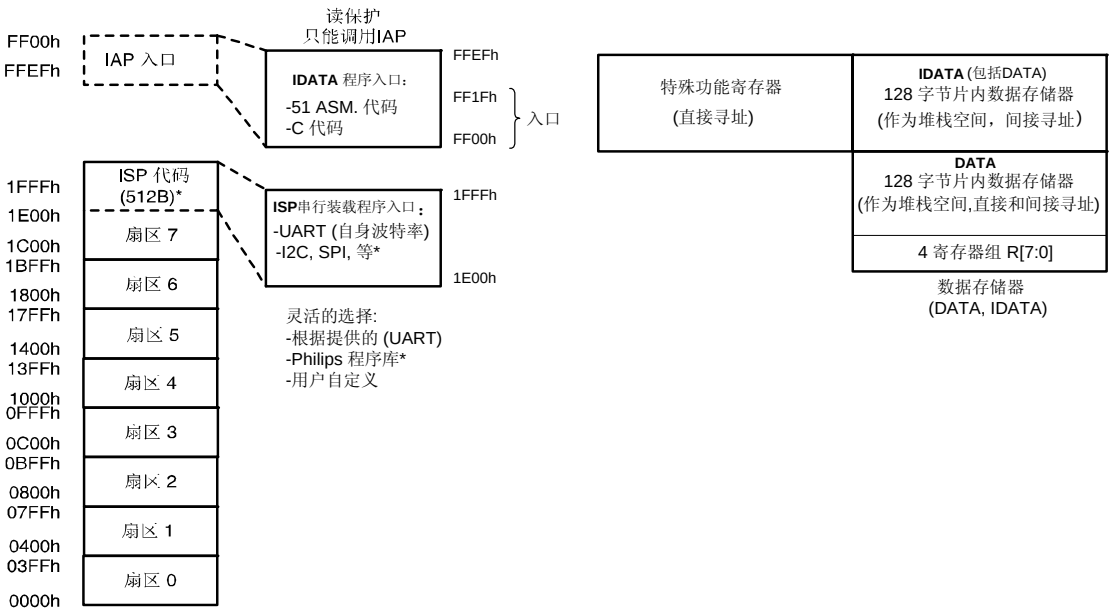


图 4 P89LPC952 存储器映射

P89LPC952 的不同存储器空间如下：

DATA －128 字节内部数据存储空间（00h:7Fh），使用除 MOVX 和 MOVC 以外的指令直接或间接寻址。可以作为全部或部分堆栈空间。

IDATA －间接数据。256 字节内部数据存储空间（00h:FFh），使用除 MOVX 和 MOVC 以外的指令间接寻址。该区域可以作为全部或部分堆栈空间，包括 DATA 和之上的 128 字节空间。

SFR －特殊功能寄存器。选中的 CPU 寄存器和外设控制及状态寄存器，只能通过直接寻址访问。

XDATA －“外部”数据或辅助 RAM。它使典型 80C51 64KB 存储空间加倍。通过 MOVX 指令使用 DPTR,R0,或 R1 进行访问。XDATA 的全部或部分空间能够在片内实现。P89LPC952 具有 512 字节片内 XDATA 存储器。

CODE －64KB 代码存储器，作为程序执行部分，通过 MOVC 指令访问。P89LPC952 具有 8KB 片内代码存储器。

表 4 数据 RAM 配置

类型	数据 RAM	字节
DATA	可直接和间接寻址的存储器	128
IDATA	间接寻址的存储器	256
XDATA	使用 MOVX,MOVC,DPTR,R0,R1 间接寻址	256

2.时钟

2.1 增强型 CPU

P89LPC952 使用增强型 80C51 CPU，其运行速度是标准 80C51 器件的 6 倍。一个机器周期由 2 个 CPU 时钟周期组成，大多数指令执行时间为 1～2 个机器周期。

2.2 时钟定义

P89LPC952 器件有几个内部时钟定义如下：

OSCCLK－输入到 DIVM 时钟分频器。OSCCLK 有 4 个时钟源选项并且也可以分频为低频时钟(见图 5 和 2.8 节"CPU 时钟(CCLK)调整：DIVM 寄存器")。**注：**fosc 定义为 OSCCLK 频率

CCLK－CPU 时钟；DIVM 时钟分频器的输出。每个机器周期包含 2 个 CCLK 周期，大多数指令执行时间为 1 到 2 个机器周期（2 到 4 个 CCLK 周期）。

RCCLK－内部 7.373MHz RC 振荡器输出。

PCLK－用于不同外围功能的时钟，为 CCLK/2。

2.2.1 振荡器时钟（OSCCLK）

LPC952 提供几个可由用户选择的振荡器选项来产生 CPU 时钟。这样就满足了从高精度到低成本的不同需求。这些选项在对 Flash 进行编程时配置，包括片内看门狗振荡器、片内 RC 振荡器、使用外部晶振的振荡器或外部时钟源。晶振可选择低、中或高频晶振，频率范围为 20KHz 到 18MHz。

2.2.2 低频振荡器选项

此选项支持 20KHz～100KHz 的外部晶振，同时也支持陶瓷谐振器。

2.2.3 中频振荡器选项

此选项支持 100KHz～4MHz 的外部晶振，同时也支持陶瓷谐振器。

2.2.4 高频振荡器选项

此选项支持 4MHz～18MHz 外部晶振，同时也支持陶瓷谐振器。**当使用频率高于 12MHz 的振荡器时，必须使能 P1.5 的复位输入功能。上电时，需要一个外部电路使器件保持复位状态，直至 V_{DD} 到达指定的电平。当系统电源被移走时，V_{DD} 将降至低于指定的最低工作电压。在某些使用频率高于 12MHz 的振荡器的应用中，当 V_{DD} 降至低于指定的最低工作电压时，需要一个外部掉电检测电路使器件保持复位状态。**

2.3 时钟输出

LPC952 支持可由用户选择的时钟输出功能。如果不使用晶振，可从 XTAL2/CLKOUT 输出时钟。实现该功能的前提是已选择另外的时钟源（片内 RC 振荡器、看门狗振荡器或 X1 脚输入的外部时钟）并且未使用晶振作为实时时钟的时钟源。这样可使外部器件与 LPC952 同步。时钟输出的使能通过置位 TRIM 寄存器中的 ENCLK 位实现。

该时钟输出的频率为 CCLK/2。如果在空闲模式中不需要输出时钟，那么可在进入空闲模式之前将该功能关闭以降低功耗。**注：**在复位时，TRIM 寄存器初始化为出厂的预编程值。当用户置位或清零 ENCLK 位时，应当保持 TRIM 寄存器其它位的内容不变。可以通过读出 TRIM 的内容（例如读入 ACC），然后修改位 6 后再写入 TRIM。另外也可使用‘ANL 直接’或‘ORL 直接’指令置位或清零 TRIM 寄存器的位 6。

2.4 片内 RC 振荡器选项

LPC952 具有一个 TRIM 寄存器，可对 RC 振荡器的频率进行调整。在复位时，TRIM 的值初始化为出厂时预编程值以将振荡器频率调整为 7.373 MHz,±1%（注:初始值误差小于 1%，请参考数据手册的温度范围内的操作）。终端用户程序可通过写 TRIM 寄存器，将片内 RC 振荡器调整为其它频率。振荡器的频率随着 TRIM 值的增加而降低。

表 5　片内 RC 振荡器 TRIM 寄存器（TRIM－地址 96h）位分配

位	7	6	5	4	3	2	1	0
符号	RCCLK	ENCLK	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0
复位	0	0	复位时，位 5:0 装入出厂编程值。					

表 6　片内 RC 振荡器 TRIM 寄存器（TRIM－地址 96h）位描述

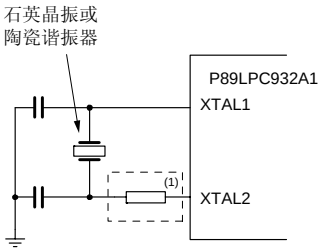
位	符号	描述
0	TRIM.0	Trim 值。决定内部 RC 振荡器的频率。复位时，将出厂的预编程值装入这几位。写该寄存器的位 6 或位 7 时，应小心处理，通过读该寄存器，按要求修改位 6 和位 7 的值并将结果装入该寄存器，这样可以保护当前的 TRIM 值。
1	TRIM.1	
2	TRIM.2	
3	TRIM.3	
4	TRIM.4	
5	TRIM.5	
6	ENCLK	当 ENCLK=1 时，若不使用晶振，XTAL2 脚输出 CCLK/2 的时钟。
7	RCCLK	当 RCCLK=1 时，选择 RC 振荡器的输出用作 CPU 时钟（CCLK）。这样，无需经过复位周期就可实现任何时钟源和内部 RC 振荡器之间的快速切换。

2.5 看门狗振荡器

看门狗有一个独立的频率为 400KHz 的振荡器。当不需要高频时钟时，可采用该振荡器来降低功耗。

2.6 外部时钟输入选项

在该配置中，处理器时钟由外部时钟源从 XTAL1/P3.1 管脚输入。频率范围为 0Hz～18MHz。XTAL2/P3.0 管脚可用作标准的 I/O 或时钟输出。**当使用频率高于 12MHz 的振荡器时，必须使能 P1.5 的复位输入功能。上电时，需要一个外部电路使器件保持复位状态，直至 V_{DD} 到达指定的电平。当系统电源被移走时，V_{DD} 将降至低于指定的最低工作电压。在某些使用频率高于 12MHz 的振荡器的应用中，当 V_{DD} 降至低于指定的最低工作电压时，需要一个外部掉电检测电路使器件保持复位状态。**



注：振荡器必须设置为下列模式之一：低频晶振，中频晶振，或高频晶振

（1）为限制晶振驱动电平可串联电阻，这对于低频晶振尤其重要（见文档）。

图 5　晶振的使用

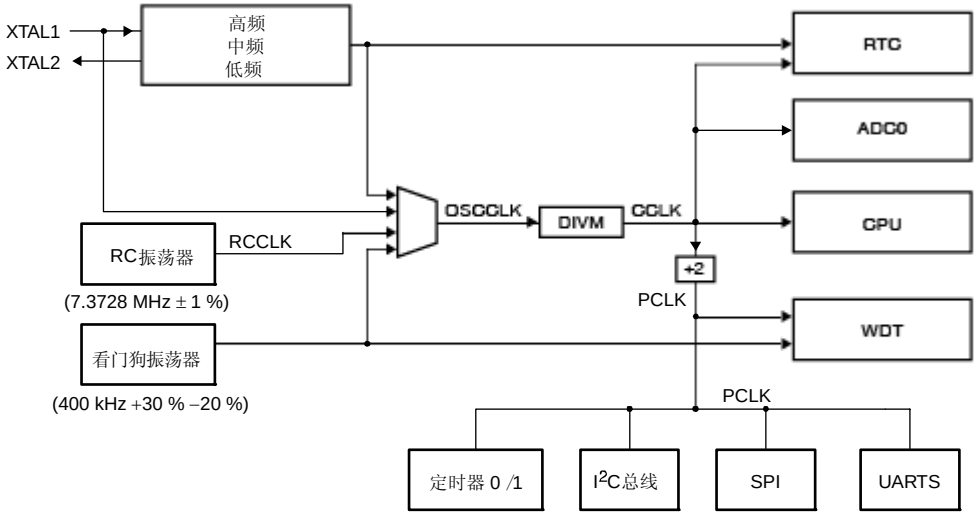


图 6 振荡器控制框图

2.7 振荡器时钟唤醒延迟

P89LPC952 有一个内部唤醒定时器，它可根据使用的时钟源来延迟时钟直到它稳定下来。如果时钟源为 3 个晶振选项之一，则延迟时间为 922 个 OSCCLK 周期加上 60～100μs，如果时钟源为内部 RC 振荡器或看门狗振荡器，则延迟时间为 224 个 OSCCLK 周期加上 60～100μs。

2.8 CPU 时钟（CCLK）调整：DIVM 寄存器

通过配置分频寄存器，OSCCLK 频率可以实现高达 510 整数分频来产生 CCLK。这样，产生的 CCLK 由下列公式获得：

CCLK 频率=f_{OSC}/(2N)

此处，f_{OSC} 为 OSCCLK 的频率，N 为 DIVM 的值。

N 的范围为 0～255，因此 CCLK 频率范围为 f_{OSC}～f_{OSC}/510。(N=0 时,CCLK=f_{OSC})。

该特性可以使 CPU 暂时在低频下运行以降低功耗。通过将时钟分频，CPU 在低频下正常执行程序，保留了对引起中断的事件以外的事件（例如，允许退出空闲模式的事件）的响应能力。这常常比空闲模式的功耗更低，与掉电模式相比则少了振荡器的启振时间。DIVM 的值可以在没有中断程序执行的情况下通过编程随时修改。

2.9 低功耗选择

P89LPC952 的 CCLK 最大时钟为 18MHz。如果 CCLK 为 8MHz 或更低，则可以置位 CLKLP SFR 位（AUXR1.7）来进一步降低功耗。在任何一次复位后，CLKLP 清零允许实现最高性能。然后，如果 CCLK 在 8MHz 或更低时钟下运行，则该位可通过软件置位。

3.A/D 转换器

3.1 概述

P89LPC952 包含 1 个 10 位、8 路逐步逼近式模数转换模块。A/D 转换器的原理框图见图 7。每个 A/D 转换器由一个 8 输入多路转换器组成。多路转换器的输出通过采样-保持电路，为两个比较器提供一个输入信号。控制逻辑连同逐次逼近式寄存器（SAR）来驱动一个数模转换器，为比较器提供另外一个输入。比较器的输出又回到 SAR。

3.2 A/D 特性

- 10 位 8 路输入的逐次逼近式 A/D 转换器。
- 8 个结果寄存器对。
- 6 种工作模式
  - －固定通道，单次转换模式
  - －固定通道，连续转换模式
  - －自动扫描，单次转换模式
  - －自动扫描，连续转换模式
  - －双通道，连续转换模式
  - －单步模式
- 3 种转换启动模式
  - －定时器触发起动

－立即起动

－边沿触发

- 在 9MHz 的 A/D 时钟下，10 位转换时间为 4us
- 中断或查询操作
- 高和低边界限制中断
- 时钟分频器
- 掉电模式

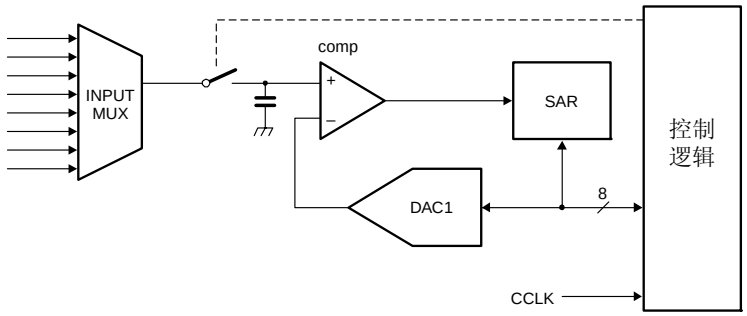


图 7　ADC 原理框图

3.2.1 A/D 工作模式

3.2.1.1 固定通道，单次转换模式

选择单个输入通道进行转换。执行单次转换并将转换结果存放到所选输入通道对应的结果寄存器中（见图 7）。如果中断使能，转换完成后产生中断。利用 ADINS 寄存器来选择输入通道。通过置位 ADMODA 寄存器的 SCAN0 位来选择该模式。

表 7　固定通道单次转换、自动扫描单次转换和自动扫描连续转换模式的输入通道和结果寄存器

结果寄存器	输入通道	结果寄存器	输入通道
AD0DAT0R/L	AD00	AD0DAT4R/L	AD04
AD0DAT1R/L	AD01	AD0DAT5R/L	AD05
AD0DAT2R/L	AD02	AD0DAT6R/L	AD06
AD0DAT3R/L	AD03	AD0DAT7R/L	AD07

3.2.1.2 固定通道，连续转换模式

选择单个通道进行连续转换。转换结果顺序存放在 8 个结果寄存器中（见表 8）。用户可选择每完成 4 次或 8 次转换后是否产生中断。其它额外的转换结果将循环存放到结果寄存器对中，将之前的结果覆盖。连续转换过程由用户终止。通过置位 ADMODA 寄存器的 SCC0 位来选择该模式。

表 8 固定通道，连续转换模式的结果寄存器和转换结果

结果寄存器	存放
AD0DAT0 R/L	所选通道，第一个转换结果
AD0DAT1 R/L	所选通道，第二个转换结果
AD0DAT2 R/L	所选通道，第三个转换结果
AD0DAT3 R/L	所选通道，第四个转换结果
AD0DAT4 R/L	所选通道，第五个转换结果
AD0DAT5 R/L	所选通道，第六个转换结果
AD0DAT6 R/L	所选通道，第七个转换结果
AD0DAT7 R/L	所选通道，第八个转换结果

3.2.1.3 自动扫描，单次转换模式

通过置位 ADINS 寄存器中相应的位来选择 8 输入通道的任意组合进行转换。执行所选输入的单次转换并将转换结果存放到所选输入通道对应的结果寄存器中（见表 7）。如果中断使能，完成前 4 次转换或所选通道的转换后产生中断。如果用户选择在 4 个输入通道转换完成后产生一个中断，那么第二个中断将在剩余的输入通道全部转换完后产生。如果只选择一个转换通道，该模式便与单通道，单次转换模式完全相同。转换按照 ADINS 寄存器 LSB 到 MSB 方向选择的通道进行。通过置位 ADMODA 寄存器的 SCAN0 位来选择该模式。

3.2.1.4 自动扫描，连续转换模式

通过置位 ADINS 寄存器的相应通道位可选择 8 输入通道的任意组合进行转换。执行所选每个通道的转换并将结果存放到转换通道相应的结果寄存器中（见表 7）。如果中断使能，完成前 4 次转换或所选通道的转换后产生中断。如果用户选择在 4 个输入通道转换完成后产生一个中断，那么第二个中断将在剩余的输入通道全部转换完后产生。自启动首次选择通道的转换开始，转换过程不断重复。其它的转换结果再循环存放到 8 个结果寄存器中，将之前的结果覆盖。连续转换过程由用户终止。转换按照 ADINS 寄存器 LSB 到 MSB 方向选择的通道进行。通过置位 ADMODA 寄存器的 BURST0 位来选择该模式。

3.2.1.5 双通道，连续转换模式

这是变化的自动扫描连续转换模式，其中转换在 2 个可用户选择的输入发生。从 8 个输入通道中选择任何 2 个通道进行转换。第一个输入通道的转换结果存放在一对结果寄存器 AD0DAT0R 和 AD0DAT0L 中。第二个输入通道的转换结果存放在结果寄存器对 AD0DAT1R 和 AD0DAT1L 中。第一个通道的第二次转换结果存放在 AD0DAT2R 和 AD0DAT2L 中，第二个通道的第二次转换结果存放在 AD0DAT3R 和 AD0DAT3L 中，如此类推。（见表 9）。如果中断使能，完成 4 次或 8 次转换后产生中断（可用户选择）。通过置位 ADMODA 寄存器的 SCC0 位来选择该模式。

表 9 双通道，连续转换模式的结果寄存器和转换结果

结果寄存器	存放
AD0DAT0 R/L	第一通道，第一个转换结果
AD0DAT1 R/L	第二通道，第一个转换结果
AD0DAT2 R/L	第一通道，第二个转换结果
AD0DAT3 R/L	第二通道，第二个转换结果

续表 9

结果寄存器	存放
AD0DAT4 R/L	第一通道，第三个转换结果
AD0DAT5 R/L	第二通道，第三个转换结果
AD0DAT6 R/L	第一通道，第四个转换结果
AD0DAT7 R/L	第二通道，第四个转换结果

3.2.1.6 单步模式

这是一种特殊的工作模式。它允许在自动扫描转换模式中进行‘单步’转换。该模式下，用户可选择 8 个输入通道的任意组合进行转换。每个通道转换结束后，产生中断（如果中断使能），A/D 转换器再等待下次转换的启动条件。每个转换结果存放在所选输入通道相应的结果寄存器中（见表 7）。该模式可与任何一种启动模式结合使用。通过清零 ADMODA 寄存器的 BURST0、SCC0 和 SCAN0 位来选择该模式。

3.2.2 转换模式选择位

A/D 转换使用 ADMODA 寄存器的 3 位来选择转换模式。这些模式位的小结见下表 10。除表中列出的组合外，3 个模式选择位的其它组合都未定义。

表 10　转换模式位

BURST0	SCC0	Scan0	ADC0 转换模式
0	0	0	单步
0	0	1	固定通道，单次转换
			自动扫描，单次转换
0	1	0	固定通道，连续转换
			双通道，连续转换
1	0	0	自动扫描，连续转换

3.2.3 转换启动模式

3.2.3.1 定时器触发启动

定时器 0 溢出时启动一次 A/D 转换。一旦转换过程被启动，其它的定时器 0 触发操作均无效，直至转换过程结束。定时器触发模式适用于所有 A/D 工作模式。该模式通过 TMMx 位、ADCS01 和 ADCS00 位来选择（见表 12 和表 14）。

3.2.3.2 立即启动

编程为该模式将立即启动一次转换。立即启动模式适用于所有 A/D 工作模式。该模式通过置位 ADCON0 寄存器的 ADCS01 和 ADCS00 位来选择（见表 12 和表 14）。

3.2.3.3 边沿触发

由 P1.4 脚的上升或下降沿来启动 A/D 转换。一旦转换过程被启动，其它的由边沿触发引起的操作均无效，直至转换过程结束。边沿触发启动模式适用于所有 A/D 工作模式。该模式通过置位 ADCON0 寄存器的 ADCS01 和 ADCS00 位来选择（见表 12 和表 14）。

3.2.4 停止和重启转换

A/D 转换的过程或转换的设置可通过清除 ADCON0 寄存器的 ADCS01 和 ADCS00 位来停止（如果转换在定时器触发模式下启动，ADCON0 寄存器的 TMM0 位也将清零）。恢复转换之前，要求用户将输入多路选择器复位到第一个用户指定的通道。这可通过写 ADINS 寄存器来实现，指定的通道对应在寄存器中的位被置位。

3.2.5 边界限制中断

每个 A/D 转换器都包含一个高、低边界限制寄存器。用户可以选择在转换结果在高和低边界限制之内（或等于高、低边界）或超出边界限制范围之外时是否产生中断。如果中断使能，转换结果超出限制范围时产生中断。边界限制通过清除边界限制中断使能来禁止。

有一个较早的检测方法可用在超出边界范围外的中断标准中。这种情况下，4 个 MSB 的转换完成后，这 4 个位与边界高和低寄存器的 4 个 MSB 相比较。如果转换的 4 个 MSB 满足中断标准的要求（即，超出边界限制范围），则产生中断（如果使能）。如果转换的 4 个 MSB 不满足中断标准的要求，当全部 8 个 MSB 转换完成后，边界限制再与之进行比较。边界状态寄存器(BNDSTA0)用来标识产生边界中断的通道。

3.2.6 时钟分频器

为了保证转换精度，A/D 转换器要求其内部时钟源的频率范围为 320kHz～9MHz。可编程时钟分频器可对时钟进行 1～8 分频（见表 16）。

3.2.7 用作 ADC 功能的 I/O 脚

模拟输入管脚可用作 I/O 口或 A/D 转换器的输入，因此也具有数据输入和输出功能。为了得到最好的模拟特性，这些用于 ADC 的管脚应当禁止其数字输出和输入功能，并且断开与 5V 最大允许电压的连接。通过将端口设置成仅为输入模式来禁止数字信号输出，详情见 I/O 口配置中描述（见表 24）。当通过置位 ADINS 寄存器中相应的位并且对应的 A/D 已被使能时，I/O 管脚的数字输入将自动断开。

当用作 I/O 口时这些管脚的电压容限为 5V。如果相应的 A/D 转换被使能且器件不处于掉电模式，则 ADINS 寄存器中选择为输入信号的管脚最大允许电压为 3V，否则这些管脚的最大允许电压仍为 5V。详细请参考 P89LPC952 数据手册。

3.2.8 掉电和空闲模式

空闲模式下，A/D 转换器（如果使能）将继续工作。如果 A/D 中断使能，转换过程结束后可使器件退出空闲模式。掉电模式和完全掉电模式下 A/D 转换器停止工作。如果 A/D 转换器被使能，它将消耗功率。因此，关闭 A/D 转换器可降低功耗。

表 11　A/D 控制寄存器 0（ADCON0-地址 97h）位分配

位	7	6	5	4	3	2	1	0
符号	ENBI0	ENADCI0	TMM0	EDGE0	ADCI0	ENADC0	ADCS01	ADCS00
复位	0	0	0	0	0	0	0	0

表 12　A/D 控制寄存器 0（ADCON0-地址 97h）位描述

位	符号	描述
1:0	ADCS01, ADCS00	A/D 启动模式位，如下：
		00－TMM0=1 时为定时器触发模式。定时器 0 溢出时启动转换。TMM0=0，不产生启动（停止模式）。 01－立即起动模式。转换立即起动。 10－边沿触发模式。当出现位 EDGE0 定义的边沿条件时转换启动。
2	ENADC0	使能 ADC0。Set＝1 时，使能 ADC0；set＝0 时，ADC 处于掉电状态。
3	ADCI0	A/D 转换结束中断 0。当任何一次转换或多个通道转换结束时该位被置位。该位由软件清零。
4	EDGE0	边沿触发模式下，该位为 0 时，由 P1.4 的下降沿来触发一次边沿转换启动；该位为 1 时，由 P1.4 的上升沿触发一次边沿转换启动。
5	TMM0	定时器触发模式 0。当 ADCS01 和 ADCS00 位＝00 时，该位用来选择停止模式（TMM0=0）或定时器触发模式（TMM0=1）。
6	ENADCI0	使能 A/D 转换结束中断 0。如果该位被置位，当 ADCI0 标志置位且 A/D 中断使能时，产生中断。
7	ENBI0	使能 A/D 边界中断 0。如果该位被置位，当边界中断 0 标志 BNDI0 置位且 A/D 中断使能时，产生中断。

表 13　A/D 模式寄存器 A（ADMODA-地址 0C0h）位分配

位	7	6	5	4	3	2	1	0
符号	BNDI0	BURST0	SCC0	SCAN0	-	-	-	-
复位	0	0	0	0	0	0	0	0

表 14　A/D 模式寄存器 A（ADMODA-地址 0C0h）位描述

位	符号	描述
0:3	-	保留
4	SCAN0	该位被置位时，选择单次转换模式（自动扫描或固定通道）。
5	SCC0	该位被置位时，选择固定和双通道，连续转换模式。
6	BURST0	该位被置位时，选择自动扫描，连续转换模式。
7	BNDI0	ADC0 边界中断标志。如果该位被置位，表明 ADC0 的转换结果超出了 ADC0 边界寄存器定义的范围。

表 15　A/D 模式寄存器 B（ADMODB-地址 A1h）位分配

位	7	6	5	4	3	2	1	0
符号	CLK2	CLK1	CLK0	INBNDO	-	-	BSA0	FCIIS
复位	0	0	0	0	0	0	0	0

表 16　A/D 模式寄存器 B（ADMODB-地址 A1h）位描述

位	符号	描述
0	FCIIS	4 个转换通道中断选择。该位为 1 时，将在固定通道或双通道连续模式的 4 次转换之后产生中断。如果选择的通道数大于 4 个，在任何一种扫描模式下置位该位都将使中断在第 4 次转换后产生。
1	BSA0	ADC0 所有边界选择。该位为 1 时，当 ADC0 的任何输入超出边界限制时 BNDI0 置位；该位为 0 时，仅当 AD00 的输入超出边界限制时 BNDI0 置位。
2:3	-	保留。
4	INBND0	该位置位时，如果转换结果在边界限制以内或等于边界限制，则产生中断。该位清零时，如果转换结果超出边界限制，则产生中断。
7:5	CLK2 CLK1 CLK0	时钟分频器，用来产生 ADC 时钟。CCLK 的分频值如下所示。分频后所得 ADC 时钟应该为 9MHz 或更低。为了保证 A/D 转换的精度，要求时钟频率不能低于 320kHz。 **CLK2:0　　分频值** 000　　1 001　　2 010　　3 011　　4 011　　5 011　　6 011　　7 011　　8

表 17　A/D 输入选择寄存器（ADINS-地址 A3h）位分配

位	7	6	5	4	3	2	1	0
符号	AIN07	AIN06	AIN05	AIN04	AIN03	AIN02	AIN01	AIN00
复位	0	0	0	0	0	0	0	0

表 18　A/D 输入选择寄存器（ADINS-地址 A3h）位描述

位	符号	描述
0	AIN00	置位时，使能 AD00 脚的采样和转换。
1	AIN01	置位时，使能 AD01 脚的采样和转换。
2	AIN02	置位时，使能 AD02 脚的采样和转换。
3	AIN03	置位时，使能 AD03 脚的采样和转换。
4	AIN04	置位时，使能 AD04 脚的采样和转换。
5	AIN05	置位时，使能 AD05 脚的采样和转换。
6	AIN06	置位时，使能 AD06 脚的采样和转换。
7	AIN07	置位时，使能 AD07 脚的采样和转换。

表 19　边界状态寄存器 0(BNDSTA0-地址 FFEDh)位分配

位	7	6	5	4	3	2	1	0
符号	BST07	BST06	BST05	BST04	BST03	BST02	BST01	BST00
复位	0	0	0	0	0	0	0	0

表 20　边界状态寄存器 0(BNDSTA0-地址 FFEDh)位描述

位	符号	描述
0	BST00	该位置位时，表示 AD00 管脚的转换结果在边界限制以内/以外。该位由软件向其写入 1 来清零。
1	BST01	该位置位时，表示 AD01 管脚的转换结果在边界限制以内/以外。该位由软件向其写入 1 来清零。
2	BST02	该位置位时，表示 AD02 管脚的转换结果在边界限制以内/以外。该位由软件向其写入 1 来清零。
3	BST03	该位置位时，表示 AD03 管脚的转换结果在边界限制以内/以外。该位由软件向其写入 1 来清零。
4	BST04	该位置位时，表示 AD04 管脚的转换结果在边界限制以内/以外。该位由软件向其写入 1 来清零。
5	BST05	该位置位时，表示 AD05 管脚的转换结果在边界限制以内/以外。该位由软件向其写入 1 来清零。
6	BST06	该位置位时，表示 AD06 管脚的转换结果在边界限制以内/以外。该位由软件向其写入 1 来清零。
7	BST07	该位置位时，表示 AD07 管脚的转换结果在边界限制以内/以外。该位由软件向其写入 1 来清零。

4 中断

P89LPC952 采用 4 优先级中断结构。这在控制 P89LPC952 的 15 个中断源的处理时提供了极大的灵活性。

任何一个中断源均可通过置位和清零中断使能寄存器 IEN0 或 IEN1 中的对应位，实现单独使能或禁止。IEN0 中还含有一个全局使能位 EA，用来使能所有中断。

任何一个中断源均可单独设置为 4 个中断优先级之一，通过置位和清零中断优先级寄存器 IP0,IP0H,IP1,和 IP1H 中的对应位来实现。正在处理的中断服务程序能够响应更高级优先级的中断，但不能响应相同或更低优先级的另一中断。最高优先级的中断不能响应其它任何中断。如果同时接收到不同优先级的两个中断请求，则响应更高优先级的请求。

如果相同优先级的中断请求在指令周期开始时同时申请中断，则内部查询序列将确定首先响应哪一个中断，这称作仲裁队列。注：仲裁队列只用于相同优先级的中断源同时申请中断的情况。表 22 汇集了了中断源，中断标志位，向量地址，使能位，优先级位，仲裁队列，以及中断是否能将 CPU 从掉电模式中唤醒等信息。

4.1 中断优先级结构

表 21　中断优先级

优先级位		中断优先级
IPxH	IPx	
0	0	0 级(最低优先级)
0	1	1 级
1	0	2 级
1	1	3 级

有 4 个特殊功能寄存器与 4 个中断优先级相关，它们分别是：IP0、IP0H、IP1 和 IP1H。每个中断都由 IPx 和 IPxH（x＝0, 1）寄存器的两个位控制，这样可将其分配到 4 个优先级中的任意一个，见表 22。

P89LPC952 有 2 个外部中断输入以及键盘中断功能。这 2 个外部中断输入和标准 80C51 微控制器相同。

通过清零或置位 TCON 寄存器中的 IT1 或 IT0，可将外部中断设为电平触发或边沿触发。如果 ITn=0，外部中断 n 通过 $\overline{\text{INTn}}$ 脚的低电平触发。如果 ITn=1，外部中断 n 为边沿触发。该模式下，对 $\overline{\text{INTn}}$ 脚连续采样，如果在一个周期为高电平而下一个周期为低电平，TCON 中的中断请求标志 IEn 将置位。这样就产生了一个中断请求。

由于外部中断脚每个机器周期采样一次，输入高或低电平应当保持至少 1 个机器周期以确保能够被正确采样到。如果外部中断为边沿触发，外部中断源应当将中断脚至少保持 1 个机器周期高电平然后至少保持 1 个机器周期低电平。这样就确保了边沿能够被检测到以使 IEn 置位。当调用中断服务程序后，CPU 自动将 IEn 清零。

如果外部中断为电平触发，外部中断源必须一直保持请求有效，直到产生所请求的中断为止。如果中断服务完成后而外部中断仍旧维持，则会产生下一次中断。当中断为电平触发时不必清除中断标志 IEn，因为中断只与输入口电平有关。

如果 P89LPC952 处于掉电或空闲状态时外部中断被编程为电平触发并且使能，中断将唤醒处理器使其恢复运行。详见 6.3 节“节电模式”。

4.2 外部中断管脚干扰抑制

P89LPC952 的大多数管脚都有干扰抑制电路，可以滤除窄脉冲的干扰（请参考 P89LPC952 数据手册的干扰滤波器的动态特性）。然而，管脚 SDA/ $\overline{\text{INT0}}$ /P1.3 和 SCL/T0/P1.2 没有干扰抑制电路。也就是说， $\overline{\text{INT1}}$ 有干扰抑制电路，而 $\overline{\text{INT0}}$ 没有干扰抑制电路。

表 22　中断汇总

描述	中断标志位	中断向量地址	中断使能位	中断优先级	仲裁顺序	掉电唤醒
外部中断 0	IE0	0003H	EX0(IEN0.0)	IP0H.0,IP0.0	1(最高)	√
定时器 0 中断	TF0	000BH	ET0(IEN0.1)	IP0H.1,IP0.1	4	
外部中断 1	IE1	0013H	EX1(IEN0.2)	IP0H.2,IP0.2	7	√
定时器 1 中断	TF1	001BH	ET1(IEN0.3)	IP0H.3,IP0.3	10	

续表 22

描述	中断标志位	中断向量地址	中断使能位	中断优先级	仲裁顺序	掉电唤醒
串口 Tx 和 Rx	TI&RI	0023H	ES/ESR (IEN0.4)	IP0H.4,IP0.4	13	
串口 Rx	RI					
掉电检测	BOF	002BH	EBO(IEN0.5)	IP0H.5,IP0.5	2	√
看门狗定时器/实时时钟	WDOVF/ RTCF	0053H	EWDRT (IEN0.6)	IP0H.6,IP0.6	3	√
I^2C 总线中断	SI	0033H	EI2C(IEN1.0)	IP0H.0,IP0.0	5	
KBI 中断	KBIF	003BH	EKBI(IEN1.1)	IP0H.0,IP0.0	8	√
比较器 1/2 中断	CMF1/CMF2	0043H	EC(IEN1.2)	IP0H.0,IP0.0	11	√
SPI 中断	SPIF	004BH	ESPI(IEN1.3)	IP1H.3, IP1.3	14	
捕获/比较单元		005BH	ECCU(IEN1.4)	IP1H.4, IP1.4	6	
串口 Tx	TI	006BH	EST(IEN1.6)	IP0H.0,IP0.0	12	
数据 EEPROM		0073H	EAD(IEN1.7)	IP1H.7, IP1.7	15	
A/D 转换器	ADCI0, BNDI1	0083H	EADC(IEN2.1)	IP2H.1,IP2.1	16(最低)	

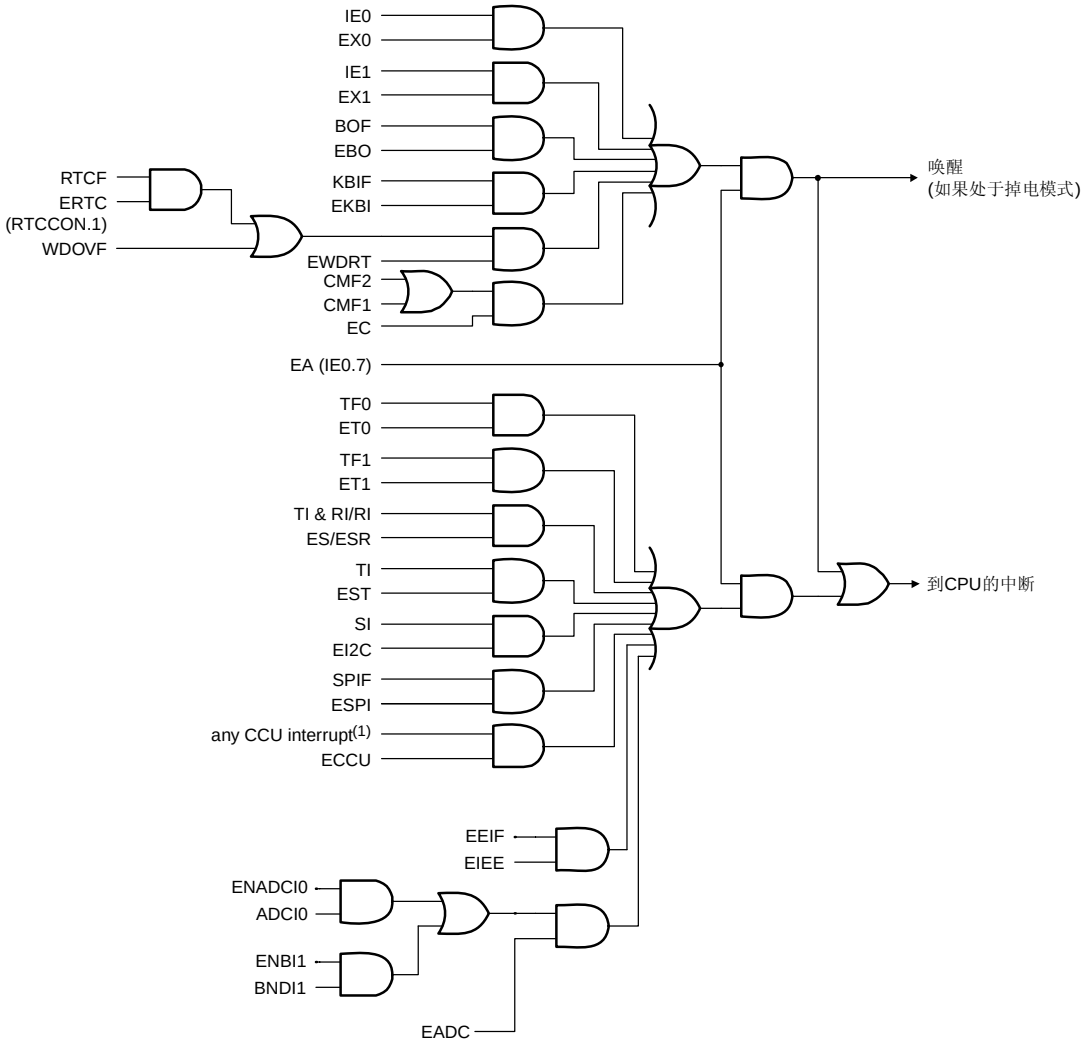


图 8 中断源，中断使能，和掉电唤醒源

5.I/O 口

P89LPC952 有 5 个 I/O 口：P0，P1，P2，P3，P4，P5。其中 P0，P1，P4，P5 为 8 位 I/O 口，P2 为 6 位 I/O 口，P3 为 2 位 I/P 口。可用的 I/O 的准确数目取决于所选的时钟和复位选项（见表 23）。

表 23 可用的 I/O 数目

时钟源	复位选项	I/O 口数目
片内振荡器或看门狗振荡器	无外部复位（上电时除外）	40
	支持外部 $\overline{\text{RST}}$ 管脚[1]	39
外部时钟输入	无外部复位（上电时除外）	39
	支持外部 $\overline{\text{RST}}$ 管脚[1]	38
低频/中频/高频振荡器（外部晶体或振荡器）	无外部复位（上电时除外）	38
	支持外部 $\overline{\text{RST}}$ 管脚[1]	37

[1] 时钟频率大于 12MHz 时必须支持外部复位方式。

5.1 端口配置

除了 3 个 I/O 管脚以外，P89LPC952 其它所有管脚都可以由软件配置为 4 种输出类型之一，如表 24 所示。这 4 种输出类型为：准双向口（标准 80C51 端口输出），推挽，开漏，仅为输入。每个端口有 2 个配置寄存器用来为端口中的管脚选择输出类型。

P1.5($\overline{\text{RST}}$)只能作为输入口，无法配置。

P1.2(SCL/T0)和 P1.3(SDA/ $\overline{\text{INT0}}$)只能配置为仅为输入或开漏。

表 24 端口输出方式设定

PxM1.Y	PxM2.Y	口输出模式
0	0	准双向口
0	1	推挽
1	0	仅为输入（高阻）
1	1	开漏

5.2 准双向口输出配置

准双向口输出类型可用作输出和输入功能而不需重新配置口线输出状态。这是因为当口线输出为逻辑高电平时驱动能力很弱，允许外部装置将其拉低。当管脚输出为低时，它的驱动能力很强，可吸收相当大的电流。准双向口有三个上拉晶体管以适应不同的需要。

在 3 个上拉晶体管中，有一个“极弱上拉”，当口线锁存为逻辑“1”时接通，当引脚悬空时，这个极弱的上拉源产生很弱的上拉电流将引脚上拉为高电平。

第 2 个上拉晶体管称为“弱”上拉，当端口锁存器为 1 且管脚本身也为 1 时接通。此上拉为输出高电平的准双向口提供主要的拉电流。如果该管脚由外部器件拉低时，弱上拉断开只有“极弱上拉”保持接通，为了把这个管脚强拉为低，外部器件必须有足够的灌电流能力使管脚上的电压降到其输入门槛电压以下。

第 3 个上拉晶体管称为“强上拉”。当口线锁存器由 0 到 1 跳变时，这个上拉用来加快准双向口上电平由逻辑 0 到逻辑 1 的转换。此时，强上拉在约 2 个机器周期内接通以使管脚能够迅速上拉到高电平。

准双向口配置如图 9 所示。

尽管 LPC952 为 3V 器件，但大多数管脚可承受 5V 电压。在准双向口模式中，如果用户在管脚加上 5V 电压，将会有电流从管脚流向 V_{DD}，这将导致额外的功率消耗。因此，建议不要在准双向口模式中向管脚施加 5V 电压。

准双向口为带干扰抑制电路的施密特触发输入。

（请参考 P89LPC952 数据手册的干扰滤波器的*动态特性*规范）。

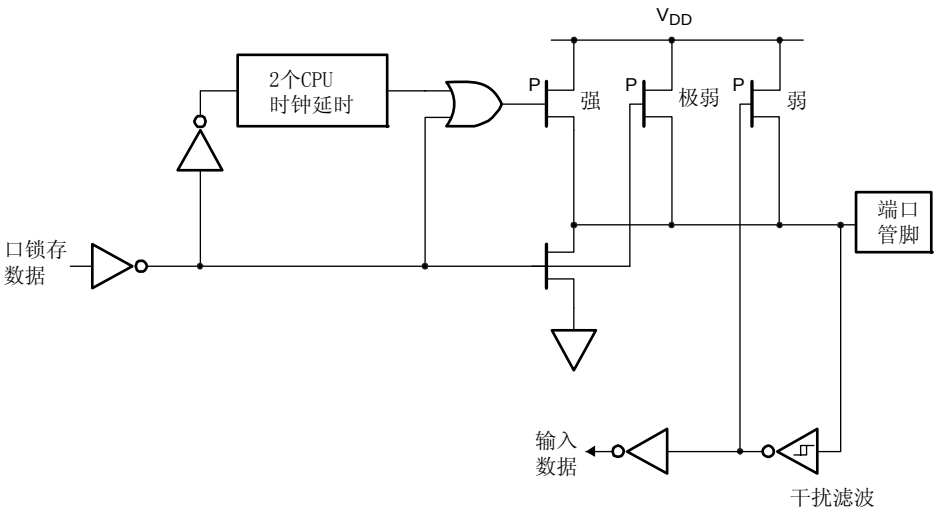


图 9　准双向口输出

5.3 开漏输出配置

当口线锁存器为‘0’时，开漏输出断开所有的上拉晶体管，仅驱动端口的下列晶体管。作为一个逻辑输出时，这种配置方式必须有外部上拉，一般通过电阻外接到 V_{DD}。这种方式的下拉和准双向口相同。

开漏输出的配置如图 10 所示。

开漏端口为带干扰抑制电路的施密特触发输入。

（请参考 P89LPC952 数据手册的干扰滤波器的*动态特性*规范）。

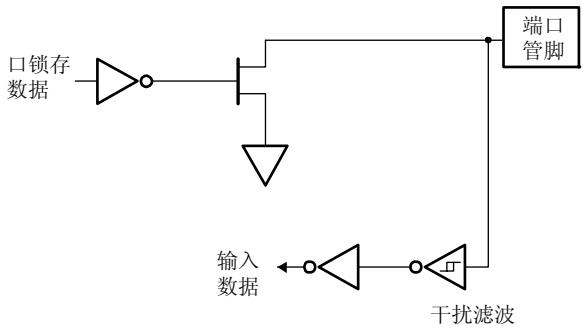


图 10　开漏输出

5.4 仅为输入配置

输入口配置如图 11 所示。输入口为带干扰抑制电路的施密特触发输入。

（请参考 P89LPC952 数据手册的干扰滤波器的*动态特性*规范）。

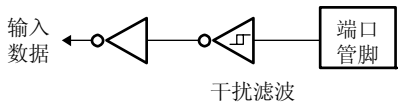


图 11　仅为输入

5.5 推挽输出配置

推挽输出配置的下拉结构与开漏输出以及准双向口的下拉结构相同，但当锁存器为‘1’时提供持续的强上拉。推挽模式一般用于输出口需要更大拉电流的情况。

推挽管脚配置如图 12 所示。

推挽管脚为带干扰抑制电路的施密特触发输入。

（请参考 P89LPC952 数据手册的干扰滤波器的*动态特性*规范）。

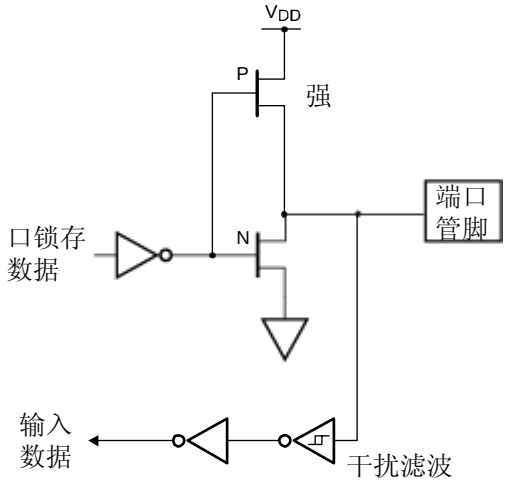


图 12　推挽输出

5.6 P0 口和模拟比较器功能

P89LPC952 集成了两个模拟比较器。为了得到最佳的模拟性能并降低功耗，用于模拟功能的管脚必须同时禁止数字输入和输出功能。

将端口设置成仅为输入时禁止数字信号输出，如端口配置部分（见图 11）所述。

P0 口的数字输入可通过 PT0AD 寄存器禁止。该寄存器的位 1-5 分别对应 P0 口管脚 P0.1-P0.5。置位 PT0AD 中的位可禁止相应管脚的数字输入。对禁止数字输入的管脚进行访问时，读出总是为 0。

任何一次复位后，PT0AD 位的 1-5 默认为 0 以使能数字功能。

5.7 附加端口特性

上电后所有的管脚都是仅为输入模式。**请注意这不同于 LPC76x 系列器件**。

- 上电之后，除 P1.5 之外，所有口都可由软件进行配置。
- P1.5 只可用于输入功能。P1.2 和 P1.3 可配置为仅为输入或开漏。

每个 LPC952 输出口都可灌入典型的 LED 驱动电流。但是所有口的输出电流总和不能超过规定的最大电流。请查阅 P89LPC952 器件手册得到详细的规格。

LPC952 所有输出端口的电平转换速度都可以控制，这就可避免因电平转换过快而导致的噪声。转换速度在出厂时设定为大约 10 ns 的上升时间和下降时间。

表 25　端口输出配置

端口管脚	配置位		可选功能	配置
	PxM1.y	**PxM2.y**		
P0.0	P0M1.0	P0M2.0	KBI0，CMP2，AD05	
P0.1	P0M1.1	P0M2.1	KBI1，CIN2B，AD00	模拟输入的用法请参考 5.6“P0 口和模拟比较器功能”
P0.2	P0M1.2	P0M2.2	KBI2，CIN2A，AD01	
P0.3	P0M1.3	P0M2.3	KBI3，CIN1B，AD02	
P0.4	P0M1.4	P0M2.4	KBI4，CIN1A，AD03	
P0.5	P0M1.5	P0M2.5	KBI5，CMPREF	
P0.6	P0M1.6	P0M2.6	KBI6，CMP1	
P0.7	P0M1.7	P0M2.7	KBI7，T1	
P1.0	P1M1.0	P1M2.0	TxD	
P1.1	P1M1.1	P1M2.1	RxD	
P1.2	P1M1.2	P1M2.2	T0，SCL	仅为输入或开漏
P1.3	P1M1.3	P1M2.3	$\overline{\textbf{INT0}}$，SDA	仅为输入或开漏
P1.4	P1M1.4	P1M2.4	$\overline{\textbf{INT1}}$	
P1.5	P1M1.5	P1M2.5	$\overline{\textbf{RST}}$	
P1.6	P1M1.6	P1M2.6	OCB	
P1.7	P1M1.7	P1M2.7	OCC，AD04	
P2.0	P2M1.0	P2M2.0	ICB，AD07	
P2.1	P2M1.1	P2M2.1	OCD，AD06	
P2.2	P2M1.2	P2M2.2	MOSI	
P2.3	P2M1.3	P2M2.3	MISO	
P2.4	P2M1.4	P2M2.4	$\overline{\textbf{SS}}$	
P2.5	P2M1.5	P2M2.5	SPICLK	
P3.0	P3M1.0	P3M2.0	CLKOUT，XTAL2	
P3.1	P3M1.1	P3M2.1	XTAL1	
P4.0	P4M1.0	P4M2.0		
P4.1	P4M1.1	P4M2.1	TROG	
P4.2	P4M1.2	P4M2.2	TxD1	

续表 25

端口管脚	配置位		可选功能	配置
	PxM1.y	PxM2.y		
P4.3	P4M1.3	P4M2.3	RxD1	
P4.4	P4M1.4	P4M2.4		
P4.5	P4M1.5	P4M2.5	SDAT	
P4.6	P4M1.6	P4M2.6		
P4.7	P4M1.7	P4M2.7	SCLK	
P5.0	P5M1.0	P5M2.0		
P5.1	P5M1.1	P5M2.1		
P5.2	P5M1.2	P5M2.2		
P5.3	P5M1.3	P5M2.3		
P5.4	P5M1.4	P5M2.4		
P5.5	P5M1.5	P5M2.5		
P5.6	P5M1.6	P5M2.6		
P5.7	P51M1.7	P5M2.7		

6.电源监控功能

P89LPC952 内含电源监控功能用于防止初始上电时的错误操作及工作时出现掉电或电源下降的情况。它通过两个硬件功能：上电检测及掉电检测（Brownout Detect）来完成。

6.1 掉电检测

掉电检测功能可用于检测电源电压是否降至某一特定值以下。掉电检测的默认操作是使处理器复位。但也可通过置位 BOI(PCON.4)位和 EBO(IEN0.5)位配置为产生一个中断。

使能和禁止掉电检测通过 BOPD(PCON.5)位，PMOD1/PMOD0(PCON[1:0])和用户配置位 BOE(UCFG1.5)来实现。如果 BOE 处于未编程状态，那么不管 PMOD1/PMOD0 和 BOPD 的状态如何，掉电检测都被禁止。如果 BOE 为已编程状态，PMOD1/PMOD0 和 BOPD 用于决定是否使能掉电检测。PMOD1/PMOD0 用于选择节电模式。如果 PMOD1/PMOD0=‘11’，掉电检测被禁止，以实现最低功耗。BOPD 默认为 0，表示如果 BOE 已编程，掉电检测将在上电时使能。

如果掉电检测使能，当 V_{DD} 低于掉电跳变电压 V_{BO}（见 P89LPC952 器件手册的*静态特性*）时产生掉电条件，并在 V_{DD} 上升超过 V_{BO} 时取消。如果 P89LPC952 器件的电源电压可以低于 2.7V，BOE 应当保持未编程状态，这样器件可在 2.4V 时工作，否则持续的掉电复位将使器件无法工作。

如果掉电检测使能（BOE 已编程，PMOD1/PMOD0≠‘11’,BOPD=0）,那么不管使能的是掉电复位还是掉电中断，BOF（RSTSRC.5）都在检测到掉电时置位。BOF 一直保持置位状态，除非通过用户软件写入“0”将其清零。需要注意的是，如果 BOE 未编程，BOF 的状态没有任何意义。如果 BOE 已编程并且发生了初始的上电，BOF 将和上电标志 POF（RSTSRC.4）一起置位。

若要正确检测到掉电，V_{DD} 上升和下降速度必须符合一定规格。请参阅数据手册。

表 26　掉电检测选项[1]

BOE (UCFG1.5)	PMOD1-0 (PCON[1:0])	BOPD (PCON.5)	BOI (PCON.4)	EBO (IEN0.5)	EA (IEN0.7)	描述
0（擦除）	XX	X	X	X	X	掉电检测禁止。V_{DD} 操作范围为 2.4V～3.6V。
1（已编程）	11(完全掉电)	X	X	X	X	
	≠11 (完全掉电模式以外的其它任何模式)	1(掉电检测关闭)	X	X	X	掉电检测禁止。V_{DD} 操作范围为 2.4V～3.6V。但在上电时 BOPD 默认为 0。
		0(掉电检测使能)	0(掉电检测产生复位)	X	X	掉电复位使能。V_{DD} 操作范围为 2.4V～3.6V。掉电检测复位时，BOF（RSTSRC.5）将置位用于指示复位源。BOF 可通过写入“0”清零。
			1(掉电检测产生中断)	1(使能掉电中断)	1(使能全局中断)	掉电中断使能。V_{DD} 操作范围为 2.4V～3.6V。掉电检测中断时，BOF（RSTSRC.5）将置位。BOF 可通过写入“0”清零。
				0	X	掉电复位和中断都被禁止。V_{DD} 操作范围为 2.4V ～ 3.6V 。但 BOF（RSTSRC.5）将在电源电压跌落到掉电检测点时置位。BOF 可通过写入“0”清零。
				X	0	

[1] 不能工作在大于 12MHz 的频率下，因为这要求 V_{DD} 在 3.0V 或 3.0V 以上。

6.2 上电检测

上电检测功能类似于掉电检测，但设计成在电源初始上电后，上升到掉电检测门槛电平之前工作。当检测到初始上电时，POF 标志(RSTSRC.4)置位。POF 将会一直保持置位状态，直到通过软件向该位写入“0”才会将其清零。需要注意的是，BOE（UCFG1.5）如果已编程，BOF（RSTSRC.5）将在 POF 置位时置位。如果 BOE 未编程，BOF 是没有意义的。

6.3 节电模式

P89LPC952 支持 3 种不同的节电模式，可通过 SFR 位 PCON[1:0]进行选择（见表 27）

表 27　节电模式

PMOD1 (PCON.1)	PMOD0 (PCON.0)	描述
0	0	正常模式(默认)—非节电模式
0	1	空闲模式。 空闲模式下片内外围功能继续工作，允许其在产生中断时激活处理器。任何一个使能的中断或复位均可结束空闲模式。
1	0	掉电模式：掉电模式将振荡器停振以使功耗最小。 掉电模式：掉电模式将振荡器停振以使功耗最小。 P89LPC952 可通过任何复位或特定的中断退出掉电模式。这些特定的中断包括－外部管脚 $\overline{INT0}$ / $\overline{INT1}$ 、掉电中断、键盘、实时时钟（系统定时器）、看门狗以及比较器触发。通过复位唤醒处理器时必须先使能相应的复位。通过中断唤醒处理器时，必须先使能相应的中断并置位 EA SFR（IEN0.7）。外部中断应当被设置成电平触发模式来退出掉电模式。 在掉电模式中，内部 RC 振荡器被禁止，除非选择 RC 振荡器作为系统时钟并使能 RTC。 在掉电模式中，电源电压可以降低到 RAM 保持电压 V_{RAM}。这样将 RAM 内容保存为进入掉电模式时的状态。SFR 内容在 V_{DD} 低于 V_{RAM} 时不受保护。因此这种情况下建议通过复位唤醒处理器。在退出掉电模式前 V_{DD} 必须上升到操作电压范围之内。 ● 当处理器从掉电模式中唤醒时，它立即启动振荡器并且当振荡器稳定下来时开始执行程序。振荡器的稳定时间是指 1024 个 CPU 时钟计数（使用其中一种晶振配置时）或 256 个时钟（使用内部 RC 或外部时钟输入配置后）。在掉电模式中某些功能继续工作并消耗电流。这样就增加了掉电时的整体功耗。这些功能包括：掉电检测 ● 看门狗定时器（当 WDCLK(WDCON.0)为“1”时） ● 比较器（注：比较器可通过置位 PCONA.5 和禁止比较器单独实现掉电） 实时时钟/系统定时器(及其所使用的振荡器电路，RTCPD(即 PCONA.7)为“1”时除外)
1	1	完全掉电模式：和掉电模式的区别在于：完全掉电模式下掉电检测电路和电压比较器都被关闭以节省功耗。注：此时不能产生掉电复位或中断。电压比较器中断和掉电中断都不可用作唤醒源。内部 RC 振荡器被禁能，除非 RC 振荡器用作系统时钟并且 RTC 被使能。 下面是所支持的唤醒源： ● 看门狗定时器(当 WDCLK(WDCON.0)为“1”时)，可以产生中断或复位来唤醒器件 ● 外部中断 $\overline{INT0}$ / $\overline{INT1}$ ● 键盘中断 ● 实时时钟/系统定时器（及晶体振荡器电路，如果模块使用，除非 RTCPD，即 PCONA.7 为“1”） 注：在掉电模式下使用内部 RC 振荡器作为 RTC 时钟源会增加相当大的功耗。当实时时钟在掉电模式下运行时，使用外部低频时钟可实现较低的功耗。

表 28　电源控制寄存器（PCON-地址 87h）位分配

位	7	6	5	4	3	2	1	0
符号	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0
复位	0	0	0	0	0	0	0	0

表 29　电源控制寄存器（PCON-地址 87h）位描述

位	符号	描述
0	PMOD0	节电模式（见 6.3 节）。
1	PMOD1	
2	GF0	通用标志 0，可通过用户软件读写，但对器件的操作并无影响。
3	GF1	通用标志 1，可通过用户软件读写，但对器件的操作并无影响。
4	BOI	掉电检测中断使能。为 1 时，检测到掉电时产生中断；为 0 时产生复位。
5	BOPD	掉电检测关闭。为 1 时，掉电检测功能处于掉电状态，即关闭状态。为 0 时，掉电检测使能。（注：在任何编程或擦除命令执行之前，BOPD 必须为 0。否则这些命令不会执行。）
6	SMOD0	帧错误位置： 为 0 时，SCON 的位 7 作为 UART 的 SM0 位被访问。 为 1 时，SCON 的位 7 作为 UART 的帧错误状态(FE)被访问
7	SMOD1	当利用定时器 1 来产生波特率时，该位为串口（UART）双倍波特率位。为 1 时，UART 的波特率为定时器 1 的溢出速率。为 0 时，UART 的波特率为定时器 1 溢出速率/2。（见第 10 节）

表 30　电源控制寄存器 A（PCONA-地址 B5h）位分配

位	7	6	5	4	3	2	1	0
符号	RTCPD	DEEPD	VCPD	ADPD	I2PD	SPPD	SPD	CCUPD
复位	0	0	0	0	0	0	0	0

表 31　电源控制寄存器 A（PCONA-地址 B5h）位描述

位	符号	描述
0	CCUPD	比较/捕获单元(CCU)掉电：该位为 1 时，CCU 的内部时钟被禁止。注:在掉电或完全掉电模式下，不管该位的状态如何，CCU 时钟都将被禁能（注: FCFG1 中的 CCUDIS 位可使该位被忽略。如果 CCUDIS=1，CCU 处于掉电模式）。
1	SPD	串口（UART）掉电：该位为 1 时，UART 的内部时钟被禁止。注：如果处于掉电或完全掉电模式下，不管该位的状态如何，UART 时钟都将会被禁止。
2	SPPD	SPI 掉电：该位为 1 时，SPI 的内部时钟被禁止。注：如果处于掉电或完全掉电模式下，不管该位的状态如何，SPI 时钟都将会被禁止。
3	I2PD	I2C 掉电：该位为 1 时，I2C 总线的内部时钟被禁止。注：如果处于掉电或完全掉电模式下，不管该位的状态如何，I2C 时钟都将会被禁止。
4	ADPD	A/D 转换器掉电：该位为 1 时，ADC 掉电。
5	VCPD	模拟电压比较器掉电：该位为 1 时，电压比较器掉电。用户必须在置位该位之前禁止电压比较器。

续表 31

位	符号	描述
6	DEEPD	数据 EEPROM 掉电：当该位为 1 时，数据 EEPROM 掉电。注：如果处于掉电或完全掉电模式下，不管该位的状态如何，数据 EEPROM 都将会掉电。
7	RTCPD	实时时钟掉电：该位为 1 时，用于实时时钟的内部时钟被禁止。

7.复位

P1.5/$\overline{\text{RST}}$ 管脚可作为低有效复位输入或数字输入口。当 UCFG1 寄存器中的位 RPE(复位管脚使能)置位时，在 P1.5 上使能外部复位输入功能。当清零时，P1.5 用作一个输入管脚。

注：在上电过程中，RPE 选择无效，该管脚总是作为外部复位输入。在上电过程中，连接到该管脚的外部电路不应将其拉低，否则将使器件一直处于复位状态。在上电完成之后，该管脚可根据 RPE 位的状态作为外部复位输入或数字输入口。只有上电复位会暂时使 RPE 的设定失效，其它复位源无法影响 RPE 位的设定。

注：为了确保上电复位的产生，电源周期内，电源使用前必须有一个 V_{DD} 下降到低于 V_{POR} 的过程（见 P89LPC952 静态特性）。

注：当使用频率高于 12MHz 的振荡器时，必须使能 P1.5 的复位输入功能。上电时，需要一个外部电路使器件保持复位状态，直至 V_{DD} 到达指定的电平。当系统电源被移走时，V_{DD} 将降至低于指定的最低工作电压。在某些使用频率高于 12MHz 的振荡器的应用中，当 V_{DD} 降至低于指定的最低工作电压时，需要一个外部掉电检测电路使器件保持复位状态。

复位可由下列复位源引起（见图 13）：

- 外部复位管脚（上电或如果用户通过 UCFG1 配置）
- 上电检测
- 掉电检测
- 看门狗定时器
- 软件复位
- UART 间隔检测复位

每一个复位源在复位寄存器 RSTSRC 中都有一个对应的标志。用户可读取该寄存器以判断最近的复位源是哪一个。这些标志位可通过软件写入“0”清零。可以有多于一个的标志位置位：

- 上电复位时，POF 和 BOF 都置位，而其它标志位清零
- 对于其它的复位，之前还没有清零的置位标志位不会受到影响

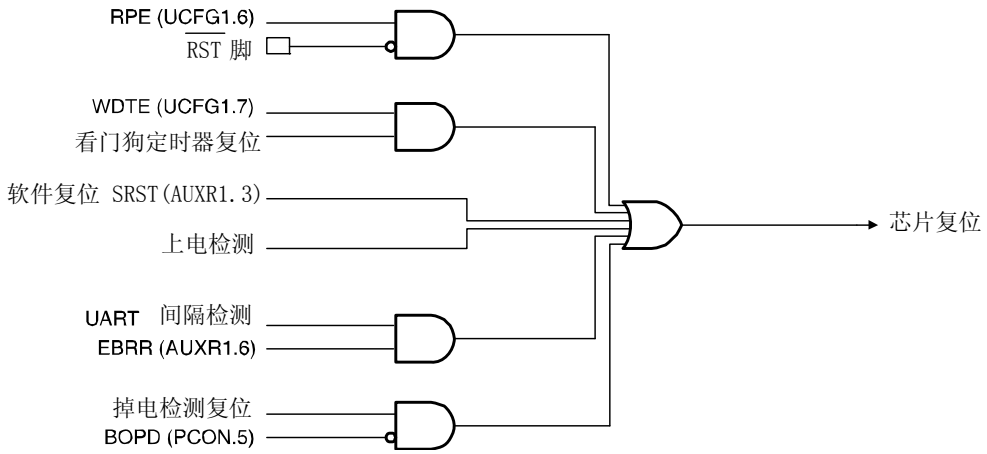


图 13 复位框图

表 32 复位源寄存器（RSTSRC-地址 DFh）位分配

位	7	6	5	4	3	2	1	0
符号	-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX
复位[1]	X	X	1	1	0	0	0	0

[1] 所示的值为上电复位的值。其它复位源将使相应的位置位。

表 33 复位源寄存器（RSTSRC-地址 DFh）位描述

位	符号	描述
0	R_EX	外部复位标志。当该位为 1 时，它指示复位源为外部管脚复位。该标志位可通过软件写入 0 或上电复位清零。如果 $\overline{\text{RST}}$ 在上电复位结束后仍然保持有效，R_EX 将置位。
1	R_SF	软件复位标志。通过软件写入 0 或上电复位清零。
2	R_WD	看门狗定时器复位标志。通过软件写入 0 或上电复位清零（注：UCFG1.7 必须为 1）。
3	R_BK	间隔检测复位。当检测到间隔并且 EBRR(AUXR1.6)置位时，产生系统复位。如果该位被置位，表明系统复位是由间隔检测引起的。通过软件写入 0 或上电复位清零。
4	POF	上电检测标志。当上电检测激活时，POF 置位以指示初始上电的状态。POF 标志将保持置位直到软件写入 0 将其清零。（注：上电复位时，BOF 和该位置位，而其它标志位都清零）。
5	BOF	掉电检测标志。当掉电检测激活时，该位置位。它将保持置位直到通过软件写入 0 将其清零。（注：上电复位时，POF 和该位置位，而其它标志位都清零）。
6:7	-	保留。

7.1 复位向量

在复位之后，P89LPC952 将从地址 0000h 或引导（Boot）地址处取指令。引导地址通过将引导向量作为地址高字节，00h 作为地址低字节构成。在发生 UART 间隔复位或非易失性引导状态位（BOOTSTAT.0）=1，或器件被强制进入 ISP 模式时，将会使用引导地址。否则，指令将从地址 0000H 处开始执行。

8.定时器／计数器 0 和 1

P89LPC952 有两个通用定时/计数器，与标准 80C51 定时器 0 及定时器 1 兼容。两者均可配置为定时器或事件计数器（见表 35）。另外增加了定时器溢出时 Tx 脚自动翻转的功能选项。

用作“定时器”功能时，每经过一个 PCLK，寄存器值加 1。

用作“计数器”功能时，寄存器在对应的外部输入管脚（T0 或 T1）上每发生一次 1 到 0 的跳变时加 1。使用该功能时，外部输入每个机器周期被采样一次。当某一周期时管脚状态采样为高而下一周期采样为低时，计数器加 1。寄存器值在检测到跳变的下一个机器周期被更新。由于检测 1 到 0 的跳变需要花费两个机器周期（4 个 CPU 时钟）的时间，所以计数频率最大值为 CPU 时钟频率的 1/4。对外部输入信号的占空比并无限制，但为了保证给定的电平信号在其改变之前至少被采样一次，信号必须至少保持一个完整的机器周期。

“定时”或“计数”功能通过特殊功能寄存器 TMOD 中的控制位 TnC/$\overline{T}$ （x＝0 和 1，分别对应定时器 0 和定时器 1）选择。定时器 0 和定时器 1 有 5 种工作模式（模式 0,1,2,3 和 6），由 TMOD 的两位（TnM1，TnM0）和 TAMOD 的 TnM2 位决定。模式 0、1、2 和 6 对于两个定时/计数器是一样的。模式 3 则不同。操作模式在后面详细描述。

表 34　定时器/计数器模式寄存器（TMOD-地址 89h）位分配

位	7	6	5	4	3	2	1	0
符号	T1GATE	T1C/$\overline{T}$	T1M1	T1M0	T0GATE	T0C/$\overline{T}$	T0M1	T0M0
复位	0	0	0	0	0	0	0	0

表 35　定时器/计数器模式寄存器（TMOD-地址 89h）位描述

位	符号	描述
0	T0M0	定时器 0 的工作模式选择。这些位和 TAMOD 寄存器的 T0M2 位一起来决定定时器 0 的工作模式（见表 37）。
1	T0M1	
2	T0C/$\overline{T}$	定时器 0 的定时或计数功能选择。该位清零时定时器 0 用作定时功能（从 CCLK 输入脉冲）；该位置位时定时器 0 用作计数功能（从 T0 脚输入脉冲）。
3	T0GATE	定时器 0 的门控位。该位被置位时，仅当 $\overline{INT0}$ 脚为高电平且 TR0 控制位被置位时使能定时器/计数器；该位被清零时，TR0 控制位被置位时使能定时器 0。
4	T1M0	定时器 1 的工作模式选择。这些位和 TAMOD 寄存器的 T1M2 位一起来决定定时器 1 的工作模式（见表 37）。
5	T1M1	
6	T1C/$\overline{T}$	定时器 1 的定时或计数功能选择。该位清零时定时器 1 用作定时功能（从 CCLK 输入脉冲）；该位置位时定时器 1 用作计数功能（从 T1 脚输入脉冲）。

续表 35

位	符号	描述
7	T1GATE	定时器 1 的门控位。该位被置位时，仅当 $\overline{\text{INT1}}$ 脚为高电平且 TR1 控制位被置位时使能定时器/计数器；该位被清零时，TR1 控制位被置位时使能定时器 1。

表 36　计数器/定时器辅助模式寄存器（TAMOD-地址 8Fh）位分配

位	7	6	5	4	3	2	1	0
符号	-	-	-	T1M2	-	-	-	T0M2
复位	x	x	x	0	x	x	x	0

表 37　计数器/定时器辅助模式寄存器（TAMOD-地址 8Fh）位描述

位	符号	描述
0	T0M2	定时器 0 的工作模式选择。这些位与 TAMOD 寄存器的位 T0M2 一起来决定定时器 0 的工作模式（见表 37）。
1:3	-	保留
4	T1M2	定时器 1 的工作模式选择。这些位与 TAMOD 寄存器的位 T1M2 一起来决定定时器 1 的工作模式（见表 37）。
		下列定时器模式由定时器模式位 TnM[2:0]选择： 000－8048 定时器“TLn”用作 5 位预分频器（模式 0）。 001－16 位定时器/计数器“THn”和“TLn”级联；无预分频器（模式 1）。 010－8 位自动重装定时器/计数器。溢出时 THn 的内容装入 TLn（模式 2）。 011－该模式下定时器 0 用作 2 个 8 位的定时器/计数器。TL0 用作一个 8 位的定时器/计数器，由标准定时器 0 控制位来控制。TH0 只可用作一个 8 位的定时器，由定时器 1 的控制位来控制（见文中描述）。该模式下定时器 1 关闭（模式 3）。 100－保留。用户不要配置成该模式。 101－保留。用户不要配置成该模式。 110－PWM 模式（见 8.5 节）。 111－保留。用户不要配置成该模式。
5:7	-	保留

8.1 模式 0

将定时器设置成模式 0 时类似 8048 定时器，即带 32 分频-预分频器的 8 位计数器。如图 14 所示。

在此模式中，定时器寄存器配置为 13 位寄存器。当计数值由全 1 变为全 0 时，置位定时器中断标志 TFn。当 TRn＝1，TnGATE=0 或 $\overline{\text{INTn}}$ ＝1 时，使能定时器的计数输入（若 TnGATE=1，定时器在外部输入 $\overline{\text{INTn}}$ 的控制下可对脉冲宽度进行测量）。TRn 是特殊功能寄存器 TCON（表 39）中的控制位。TnGATE 是 TMOD 寄存器的位。

该 13 位寄存器由 THn 的 8 位和 TLn 的低 5 位组成。TLn 的高 3 位不定，可忽略。置位运行标志（TRn）不会清零寄存器。

定时器 0 和定时器 1 的模式 0 操作相同，见图 14。有两个不同的 GATE 位，分别对应定时器 1 （TMOD.7）和定时器 0（TMOD.3）。

8.2 模式 1

模式 1 除了使用的是 16 位的定时器寄存器（THn 和 TLn）外，其它与模式 0 相同，见图 15。

8.3 模式 2

在此模式中，定时器寄存器作为可自动重装的 8 位计数器（TLn），如图 16 所示。TLn 溢出不仅置位 TFn，而且将 THn 的值重新装入 TLn（THn 的值由软件预置）。重装时 THn 不变。定时器 0 及定时器 1 在模式 2 中的操作相同。

8.4 模式 3

在模式 3 中定时器 1 停止工作。等效于使 TR1=0。

在此模式中，定时器 0 的 TL0 及 TH0 作为两个独立的 8 位计数器。定时器 0 模式 3 的逻辑如图 17 所示。TL0 使用定时器 0 的控制位：T0C/$\overline{\text{T}}$,T0GATE,TR0, $\overline{\text{INT0}}$ 和 TF0。TH0 只用作定时功能（计数机器周期），占用定时器 1 的 TR1 和 TF1。因此，该模式下 TH0 控制“定时器 1”的中断。

模式 3 用于需要一个额外的 8 位定时器的场合。当定时器 0 处于模式 3 时，P89LPC952 可以看成是拥有 3 个定时器/计数器的器件。

注：当定时器 0 工作在模式 3 时，定时器 1 可通过进入/退出其自身的模式 3 来起动/关闭。它仍可用作串口的波特率发生器，或应用在不需要中断的任何场合。

8.5 模式 6

在该模式中，相应的定时器可以改变为周期为 256 个定时器时钟的 PWM（见图 18）。除以下几点外，其结构与模式 2 相同：

- TFn（n＝0 和 1，分别对应定时器 0 和 1）由硬件置位和清零；

- TFn 的低电平时间为 THn 的值，必须在 1～254 之间，并且：
- TFn 的高电平时间为 256-THn。
- 向 THn 装入 00h 使 Tx 管脚强制为高电平，装入 FFh 使 Tx 管脚强制为低电平。

注：TFn 的由低到高跳变时中断仍可以使能，并且这个 TFn 可象其它模式一样仍由软件清零。

表 38　定时器/计数器控制寄存器（TCON-地址 88h）位分配

位	7	6	5	4	3	2	1	0
符号	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
复位	0	0	0	0	0	0	0	0

表 39　定时器/计数器控制寄存器（TCON-地址 88h）位描述

位	符号	描述
0	IT0	中断 0 类型控制位。由软件置位/清零以选择外部中断以下降沿/低电平方式触发。
1	IE0	中断 0 边沿标志。当检测到外部中断 0 边沿时由硬件置位该标志。中断处理时由硬件清零，或通过软件清零。
2	IT1	中断 1 类型控制位，由软件置位/清零以选择外部中断以下降沿/低电平方式触发。
3	IE1	中断 1 边沿标志。当检测到外部中断 1 边沿时由硬件置位该标志。中断处理时由硬件清零，或通过软件清零。
4	TR0	定时器 0 运行控制位。通过软件将其置位/清零来起动/关闭定时器/计数器 0。
5	TF0	定时器 0 溢出标志。定时器/计数器溢出时由硬件置位。中断执行时硬件清零，或由软件清零（模式 6 除外，在模式 6 中由硬件清零）。
6	TR1	定时器 1 运行控制位。通过软件将其置位/清零来起动/关闭定时器/计数器 1。
7	TF1	定时器 1 溢出标志。定时器/计数器溢出时由硬件置位。中断执行时硬件清零，或由软件清零（模式 6 除外，在模式 6 中由硬件清零，见以上有关模式 6 的描述）。

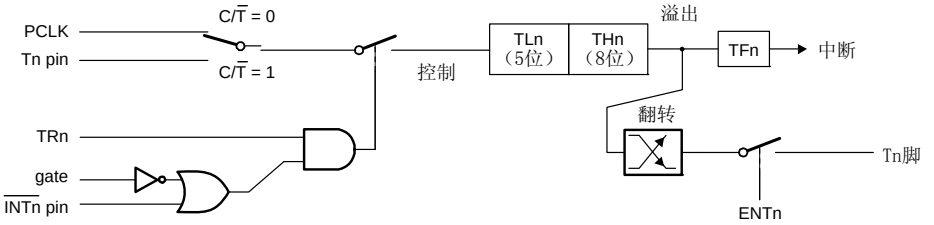


图 14　定时器/计数器 0 或 1 的工作模式 0（13 位计数器）

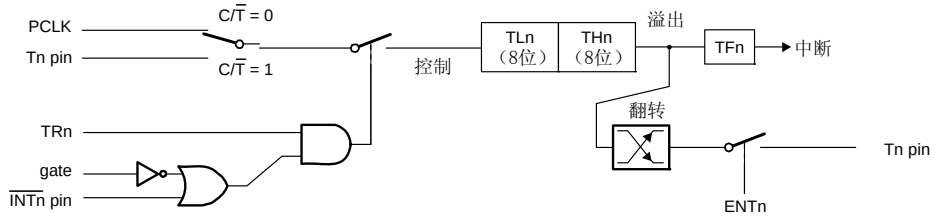


图 15　定时器/计数器 0 或 1 的工作模式 1（16 位计数器）

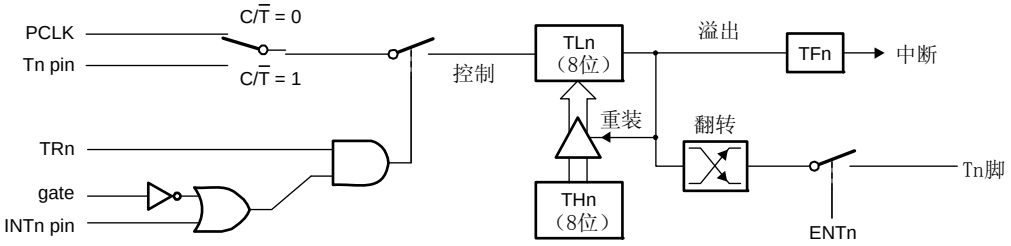


图 16　定时器/计数器 0 或 1 的工作模式 2（8 位自动重装）

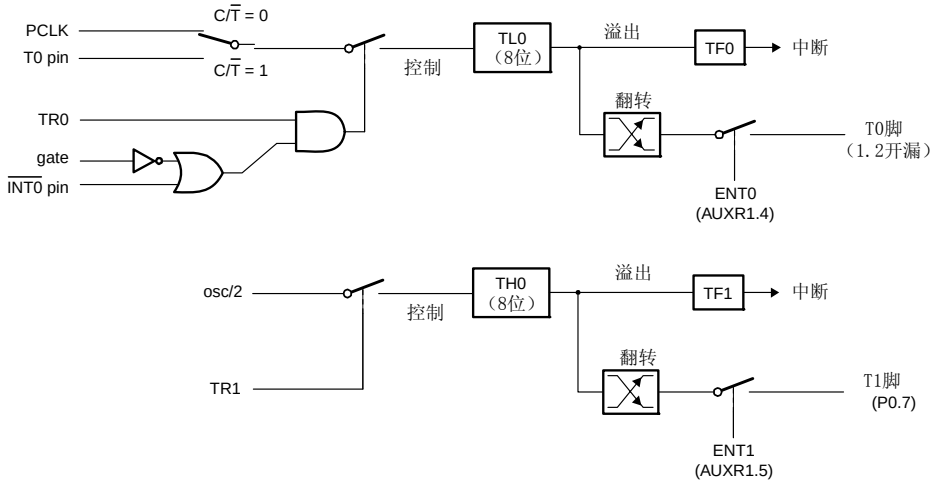


图 17　定时器/计数器 0 的工作模式 3（2 个 8 位计数器）

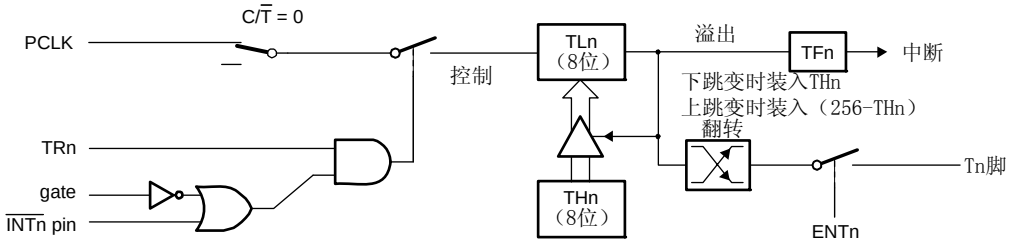


图 18　定时器/计数器 0 或 1 的工作模式 6（PWM 自动重装）

8.6 定时器溢出触发输出

定时器 0 和 1 可配置为发生定时器溢出时自动触发端口输出。T0/T1 的计数输入和 PWM 输出占用相同的管脚，也可用作定时器触发输出。此功能通过配置 AUXR1 寄存器的 ENT0 和 ENT1 控制位来实现，并分别应用到定时器 0 或定时器 1。打开该模式后，在首次定时器溢出之前端口的输出为逻辑 1。为了实现该功能，C/$\overline{T}$ 必须清零来选择 PCLK 作为定时器的时钟源。

9.实时时钟系统定时器

P89LPC952 具有一个简单的实时时钟/系统定时器。它允许用户在器件其它部分掉电时能够继续运行一个精确的定时器。实时时钟可以作为一个中断或一个唤醒源（见图 19）。

实时时钟是一个 23 位的倒计数器。该计数器的时钟源可以是 CPU 时钟（CCLK）或者 XTAL1-2 振荡器（前提是 XTAL1-2 振荡器不作为 CPU 的时钟源）。如果 XTAL1-2 振荡器作为 CPU 时钟源，不管 RTCCON 寄存器的 RTCS1:0 状态如何，RTC 将使用 CCLK 作为它的时钟源。RTC 有 3 个相关的 SFR：

- RTCCON－实时时钟控制。
- RTCH－实时时钟计数器重装高字节（位[22:15]）。
- RTCL－实时时钟计数器重装低字节（位[14:7]）。

实时时钟/系统定时器通过置位 RTCEN（RTCCON.0）使能。实时时钟是一个 23 位的倒计数器（当 RTCEN=0 时初始化成全‘0’），它由 1 个 7 位的预分频器和 1 个 16 位的可装载倒计数器组成。当 RTCEN 写入‘1’时，计数器先装载（RTCH，RTCL,‘1111111’），然后向下计数。当它为全‘0’时，计数器再次重装（RTCH,RTCL,‘1111111’），并建立 RTCF（RTCCON.7）标志。

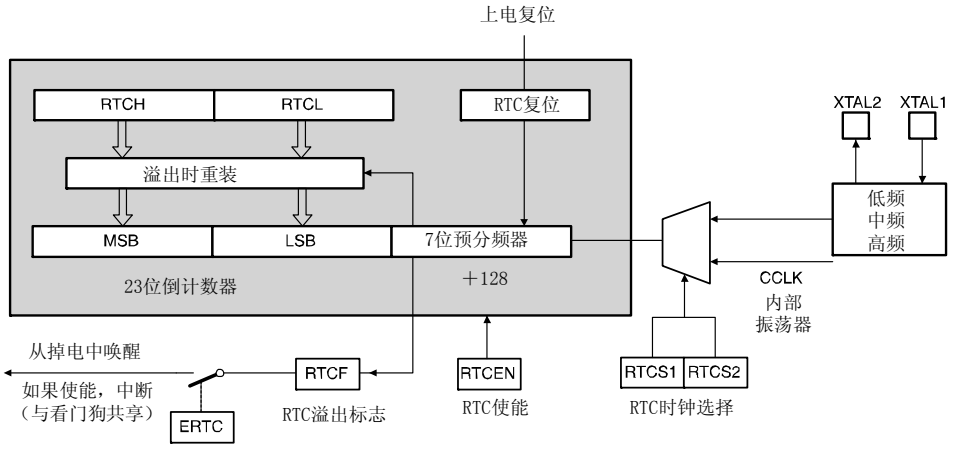


图 19　实时时钟/系统定时器框图

9.1 实时时钟源

如果内部 RC 振荡器或内部看门狗振荡器用作 CPU 时钟，通过配置 RTCS1/RTCS0（RTCCON[6:5]）来选择 RTC 的时钟源。如果内部晶振或由 XTAL1 输入的外部时钟作为 CPU 时钟，RTC 将使用 CCLK 作为它的时钟源。

9.2 RTCS1/RTCS0 的更改

若 RTC 当前使能（即 RTCCON.0=1）时，RTCS1/RTCS0 不能被改变。置位 RTCEN 和更改 RTCS1/RTCS0 可以在一条对 RTCCON 的写指令内完成。但如果 RTCEN=1，在更新 RTCS1/RTCS0 之前必须先清零 RTCEN。

9.3 实时时钟中断/唤醒

如果 ERTC(RTCCON.1), EWDRT(IEN1.0.6)和 EA(IEN0.7)置位为 1，RTCF 可作为一个中断源。该中断向量和看门狗定时器的中断向量相同。它也可以作为器件的一个唤醒源。

9.4 影响实时时钟的复位源

只有上电复位会将实时时钟及其相关的 SFR 复位为默认状态。

表 40　实时时钟/系统定时器时钟源

FOSC2:0	RCCLK	RTCS1:0	RTC 时钟源	CPU 时钟（CCLK）
000	0	00	高频晶振	高频晶振/DIVM
		01		
		10		
		11	高频晶振/DIVM	
000	1	00	高频晶振	内部 RC 振荡器
		01		
		10		
		11	内部 RC 振荡器	
001	0	00	中频晶振	中频晶振/DIVM
		01		
		10		
		11	中频晶振/DIVM	
001	1	00	中频晶振	内部 RC 振荡器
		01		
		10		
		11	内部 RC 振荡器	
010	0	00	低频晶振	低频晶振/DIVM
		01		
		10		
		11	低频晶振/DIVM	
010	1	00	低频晶振	内部 RC 振荡器
		01		
		10		
		11	内部 RC 振荡器	
011	0	00	高频晶振	内部 RC 振荡器/DIVM
		01	中频晶振	
		10	低频晶振	
		11	内部 RC 振荡器/DIVM	

续表 40

FOSC2:0	RCCLK	RTCS1:0	RTC 时钟源	CPU 时钟（CCLK）
011	1	00	高频晶振	内部 RC 振荡器
		01	中频晶振	
		10	低频晶振	
		11	内部 RC 振荡器	
100	0	00	高频晶振	看门狗振荡器/DIVM
		01	中频晶振	
		10	低频晶振	
		11	看门狗振荡器/DIVM	
100	1	00	高频晶振	内部 RC 振荡器
		01	中频晶振	
		10	低频晶振	
		11	内部 RC 振荡器	
101	X	XX	未定义	未定义
110	X	XX	未定义	未定义
111	0	00	外部时钟输入	外部时钟输入/DIVM
		01		
		10		
		11	外部时钟输入/DIVM	
111	1	00	外部时钟输入	内部 RC 振荡器
		01		
		10		
		11	内部 RC 振荡器	

表 41　实时时钟控制寄存器（RTCCON-地址 D1h）位分配

位	7	6	5	4	3	2	1	0
符号	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN
复位	0	1	1	x	x	x	0	0

表 42　实时时钟控制寄存器（RTCCON-地址 D1h）位描述

位	符号	描述
0	RTCEN	实时时钟使能。如果该位为 1，实时时钟使能。注：该位不会使实时时钟进入掉电状态。不管 RTCEN 的状态如何，RTCPD (PCONA.7)为 1 时会使实时时钟进入掉电，并且禁止该模块。
1	ERTC	实时时钟中断使能。实时时钟和看门狗定时器共用一个中断源。注：如果用户配置位 WDTE(UCFG1.7) 为 0，看门狗定时器可使能来产生中断。用户可读取 RTCF(RTCCON.7)位以确定中断是否由实时时钟产生。
2:4	-	保留
5	RTCS0	实时时钟源选择（见表 40）
6	RTCS1	
7	RTCF	实时时钟标志。当 23 位实时时钟到达计数值 0 时，该位置位。该位可通过软件清零。

10.URAT

P89LPC952 具有两个增强型的 UART，与传统的 80C51 UART 兼容，但有一点除外，即定时器 2 的溢出不能用于产生波特率。P89LPC952 每个 UART 都还带有一个独立的波特率发生器。波特率可以选择由振荡器（经常数分频），定时器 1 溢出或者独立的波特率发生器产生。除了产生波特率以外，在标准 80C51 UART 基础上还增加了帧错误检测、间隔检测、自动地址识别、可选的双缓冲以及几个中断选项。

UART 具有 4 种工作模式，分别论述如下：

10.1 模式 0

串行数据通过 RxDn 进出。TxDn 输出移位时钟。每次发送或接收都为 8 位，LSB（最低位）在前。波特率固定为 CPU 时钟频率的 1/16。

10.2 模式 1

TxDn 脚发送，RxDn 脚接收，每次数据为 10 位：1 个起始位（逻辑 0），8 个数据位（LSB 在前）以及 1 个停止位（逻辑 1）。当接收数据时，停止位保存于特殊功能寄存器 SnCON 的 RB8_n 中。该模式的波特率可变，由定时器 1 溢出速率或波特率发生器决定（详见 10.6 节“波特率发生器及其选择”）。

10.3 模式 2

TxDn 脚发送，RxDn 脚接收，每次数据为 11 位：1 个起始位（逻辑 0），8 个数据位（LSB 在前），一个可编程第 9 位数据及 1 个停止位（逻辑 1）。发送数据时，第 9 个数据位（SnCON 中的 TB8 位）可置为 0 或 1。例如可将奇偶位（PSW 内 P 位）放入 TB8。接收时，第 9 位数据存入 SnCON 的 RB8_n 位，而停止位不会被保存。波特率可编程为 CCLK 频率的 1/16 或 1/32，由 PCON 内 SMOD1 位决定，SMOD1 位用于两个 UART。

10.4 模式 3

TxDn 脚发送，RxDn 脚接收，每次数据为 11 位：1 个起始位（逻辑 0），8 个数据位（LSB 在前），1 个可编程的第 9 位数据及 1 个停止位（逻辑 1）。实际上，模式 3 除了波特率外其它均与模式 2 相同。模式 3 的波特率可变并由定时器 1 溢出率或波特率发生器决定（详见 10.6 节“波特率发生器及其选择”）。

在上述 4 种模式中，发送过程是以一条 SnBUF 作为目标寄存器的指令开始的。在模式 0 中接收过程通过设置 R1_n=0 及 REN_n=1 启动，在其它模式中，如果 REN_n=1 则通过接收起始位来启动。

10.5 SFR 空间

UART 的 SFR 位于以下位置：

表 43　UART 的 SFR 地址

寄存器	描述	SFR 位置
PCON	电源控制	87H
S0CON	串口（UART0）控制	98H
S0BUF	串口（UART0）数据缓冲区	99H
S0ADDR	串口（UART0）地址	A9H
S0ADEN	串口（UART0）地址使能	B9H
S0STAT	串口（UART0）状态	BAH
BRGR1_0	波特率发生器 0 高字节	BFH
BRGR0_0	波特率发生器 0 低字节	BEH
BRGCON_0	波特率发生器 0 控制	BDH
S1CON	串口（UART）控制	B6H
S1BUF	串口（UART）数据缓冲区	FFB0H
S1ADDR	串口（UART）地址	FFB2H
S1ADEN	串口（UART）地址使能	FFB1H
S1STAT	串口（UART）状态	D4H
BRGR1_1	波特率发生器高字节	FFB5H
BRGR0_1	波特率发生器低字节	FFB4H
BRGCON_1	波特率发生器控制	FFB3H

10.6 波特率发生器及选择

P89LPC952 的增强型 UART 具有一个独立的波特率发生器。波特率取决于对 BRGR1_n 和 BRGR0_n 编程的值。每个 UART 都可使用定时器 1 或者波特率发生器的输出来产生波特率，由 BRGCON[2:1]决定（见图 20）。需要注意的是，如果 SMOD1(PCON.7)置位，定时器 T1 被 2 分频（用于两个 UART）。独立的波特率发生器使用 CCLK 作为时钟源。

10.7 更新 BRGR1 和 BRGR0 SFR 寄存器

波特率 SFR 寄存器，BRGR1_n 和 BRGR0_n 只能在波特率发生器禁止（BRGCON 寄存器的 BRGEN_n 位为‘0’）时写入。这样避免了向波特率发生器装入临时的值（**警告：当 BRGEN=1 时写入 BRGR0_n 或 BRGR1_n，结果将是不可预知的**）。

表 44　UART 的波特率产生

SCON.7 (SM0)	SCON.6 (SM1)	PCON.7 (SMOD1)	BRGCON.1 (SBRGS)	UART 接收/发送波特率
0	0	X	X	CCLK/16
0	1	0	0	CCLK/(256－TH1)64
		1	0	CCLK/(256－TH1)32
		X	1	CCLK/((BRGR1_n, BRGR0_n)+16)
1	0	0	X	CCLK/32
		1	X	CCLK/16
1	1	0	0	CCLK/(256－TH1)64
		1	0	CCLK/(256－TH1)32
		X	1	CCLK/((BRGR1_n, BRGR0_n)+16)

表 45　波特率发生器控制寄存器（BRGCON_0-地址 BDh）位分配

位	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	-	SBRGS_0	BRGEN_0
复位	x	x	x	x	x	x	0	0

表 46　波特率发生器控制寄存器（BRGCON_0-地址 BDh）位描述

位	符号	描述
0	BRGEN_0	波特率发生器 0 使能。只有当 BRGEN_0 为 0 时，才可对寄存器 BRGR1_0 和 BRGR0_0 进行写操作。
1	SBRGS_0	选择波特率发生器 0 用于产生 UART0 模式 1 和 3 的波特率（详见表 44）
2:7	-	保留

表 47　波特率发生器控制寄存器（BRGCON_1-地址 FFB3h）位分配

位	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	-	SBRGS_1	BRGEN_1
复位	x	x	x	x	x	x	0	0

表 48　波特率发生器控制寄存器（BRGCON_1-地址 FFB3h）位描述

位	符号	描述
0	BRGEN_1	波特率发生器 1 使能。只有当 BRGEN_1 为 0 时，才可对寄存器 BRGR1_1 和 BRGR0_1 进行写操作。
1	SBRGS_1	选择波特率发生器 0 用于产生 UART1 模式 1 和 3 的波特率（详见表 44）
2:7	-	保留

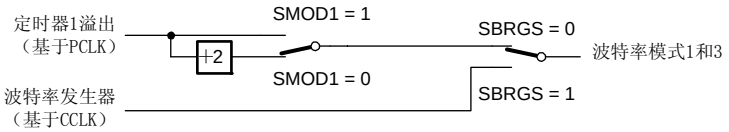


图 20　UART 波特率的产生（模式 1，3）

10.8 帧错误

当检测到停止位为 0 时产生帧错误。帧错误在状态寄存器（SnSTAT）中报告。此外，如果 SMOD0(PCON.6)为 1，SnCON.7 可作为帧错误位。如果 SMOD0 为 0，SCON.7 作为 SM0_0 以及 S1CON 作为 SM0_1。建议在 SMOD0 为 0 时对 SM0_n 和 SM1_n(SnCON[7:6])进行设置。

10.9 间隔检测

间隔检测在状态寄存器（SnSTAT）中报告。当连续检测到 11 个位都为低电平位时，则认为检测到一个间隔。由于间隔条件同样满足帧错误条件，因此检测到间隔时也会报告帧错误。一旦检测到间隔条件，UART 将进入空闲模式并一直保持，直至接收到停止位。UART0 间隔检测还可用于对器件复位并通过置位 EBRR 位（AUXR1.6）使器件进入 ISP 模式。UART1 间隔检测不能使器件复位，但能用来产生一个中断。

表 49　串口 0 控制寄存器（S0CON-地址 98h）位分配

位	7	6	5	4	3	2	1	0
符号	SM0_0/FE_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0
复位	x	x	x	x	x	x	0	0

表 50　串口 0 控制寄存器（S0CON-地址 98h）位描述

位	符号	描述
0	RI_0	接收中断标志 0。在模式 0 中，第 8 位接收完成时由硬件置位。模式 1 中，大约在停止位的中间时刻由硬件置位。模式 2 或 3 中，如果 SMOD0=0，在接近第 9 位（位 8）的中间时刻置位；如果 SMOD0=1，在接近停止位的中间时刻置位（例外见 SM2_0-S0CON.5）。必须通过软件清零。
1	TI_0	发送中断标志 0。在模式 0 中，在第 8 位数据发送结束时由硬件置位。在其它模式中，在发送停止位（见 S0STAT 寄存器中 INTLO_0 位的描述）时由硬件置位。必须通过软件清零。
2	RB8_0	在模式 2 和 3 中，该位为接收到的第 9 位数据，在模式 1 中（SM2_0 必须为 0），为接收到的停止位。在模式 0 中，RB8_0 未定义。
3	TB8_0	模式 2 和 3 中将要发送的第 9 位数据，可以根据需要由软件置位或清零。
4	REN_0	使能串行接收。由软件置位以使能接收。软件清零则禁止接收。
5	SM2_0	使能模式 2 和 3 中的多机通信功能。在模式 2 或 3 中，如果 SM2_0=1 而且接收到的第 9 位数据（RB8_0）为 0 时，则 RI_0 不会被激活。在模式 0 中，SM2_0 应当为 0。在模式 1 中，SM2_0 必须为 0。
6	SM1_0	和 SM0_1 定义串行口操作模式（见表 53）

续表 50

位	符号	描述
7	SM0_0/FE_0	该位的用途由 PCON 寄存器中的 SMOD0 决定。如果 SMOD0=0，该位作为 SM0_0 读和写，和 SM1_0 一起定义串口模式；如果 SMOD0=1，该位作为 FE_0(帧错误)。当检测到一个无效的停止位时，FE 被接收器置位。一旦置位后，该位不能由有效帧清零，但可以通过软件清零。（注：UART0 模式位 SM0_0 和 SM1_0 应当在 SMOD0=0（任何复位后默认的状态）时进行设置。）

表 51　串口 1 控制寄存器（S1CON-地址 B6h）位分配

位	7	6	5	4	3	2	1	0
符号	SM0_1/FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1
复位	x	x	x	x	x	x	0	0

表 52　串口 1 控制寄存器（S1CON-地址 B6h）位描述

位	符号	描述
0	RI_1	接收中断标志 1。在模式 0 中，第 8 位接收完成时由硬件置位。模式 1 中，大约在停止位的中间时刻由硬件置位。模式 2 或 3 中，如果 SMOD0=0，在接近第 9 位（位 8）的中间时刻置位；如果 SMOD0=1，在接近停止位的中间时刻置位（例外见 SM2_1-S1 CON.5）。必须通过软件清零。
1	TI_1	发送中断标志 1。在模式 0 中，在第 8 位数据发送结束时由硬件置位。在其它模式中，在发送停止位（见 S1STAT 寄存器中 INTLO_1 位的描述）时由硬件置位。必须通过软件清零。
2	RB8_1	在模式 2 和 3 中，该位为接收到的第 9 位数据，在模式 1 中（SM2_1 必须为 0），为接收到的停止位。在模式 0 中，RB8_1 未定义。
3	TB8_1	模式 2 和 3 中将要发送的第 9 位数据，可以根据需要由软件置位或清零。
4	REN_1	使能串行接收。由软件置位以使能接收。软件清零则禁止接收。
5	SM2_1	使能模式 2 和 3 中的多机通信功能。在模式 2 或 3 中，如果 SM2_1=1 而且接收到的第 9 位数据（RB8_1）为 0 时，则 RI_1 不会被激活。在模式 0 中，SM2_1 应当为 0。在模式 1 中，SM2_1 必须为 0。
6	SM1_1	和 SM0_1 定义串行口操作模式（见表 53）
7	SM0_1/FE_1	该位的用途由 PCON 寄存器中的 SMOD0 决定。如果 SMOD0=0，该位作为 SM0_1 读和写，和 SM1_1 一起定义串口模式；如果 SMOD0=1，该位作为 FE_1(帧错误)。当检测到一个无效的停止位时，FE 被接收器置位。一旦置位后，该位不能由有效帧清零，但可以通过软件清零。（注：UART1 模式位 SM0_1 和 SM1_1 应当在 SMOD0=0（任何复位后默认的状态）时进行设置。）

表 53　串口模式

SM0_n, SM1_n	UART 模式	UART 波特率
00	模式 0：移位寄存器	CCLK/16（任何复位后的默认模式）
01	模式 1：8 位 UART	变量（见表 44）
10	模式 2：9 位 UART	CCLK/32 或 CCLK/16
11	模式 3：9 位 UART	变量（见表 44）

表 54　串口 0 状态寄存器（S0STAT-地址 BAh）位分配

位	7	6	5	4	3	2	1	0
符号	DBMOD_0	INTLO_0	CIDIS_0	DBISEL_0	FE_0	BR_0	OE_0	STINT_0
复位	x	x	x	x	x	x	0	0

表 55　串口 0 状态寄存器（S0STAT-地址 BAh）位描述

位	符号	描述
0	STINT_0	状态中断使能 0。该位置位时，FE_0, BR_0 或 OE_0 可产生中断。这些中断（向量地址 0023h）与 RI（CIDIS=1 时）或组合 TI/RI（CIDIS=0）共用中断向量。该位清零时，FE_0, BR_0，OE_0 不产生中断。（注：FE_0, BR_0 或 OE_0 位通常和 RI_0 一起产生，不管 STINT 的状态如何，RI_0 都将产生中断。）注：如果 EBRR(AUXR1.6)置位，BR_0 可导致间隔检测复位。
1	OE_0	当接收缓冲区仍然为满时接收到一个新的字符（在软件读出缓冲区中的前一个字符之前），也就是当 RI_0 仍然为置位状态时接收到新字节的位 8，那么溢出错误 0 标志 OE_0 将置位。该位通过软件清零。
2	BR_0	间隔检测 0 标志。当连续检测到 11 个位都为低电平时表明检测到一个间隔。该位通过软件清零。
3	FE_0	帧错误 0 标志。当接收器在数据帧结束时没有收到有效的停止位时，该标志置位。通过软件清零。
4	DBISEL_0	双缓冲发送中断选择 0。只在双缓冲使能时使用。该位在双缓冲使能时控制产生中断的次数。当该位置位时，每向 S0BUF 写入一个字符后产生一个发送中断，另外在最后一个字符发送的停止位（即缓冲区已空）开始（INTLO_0=0）或结束（INTLO_0=1）时再产生一次发送中断。最后这次中断可用于指示所有发送操作已完成。当该位为 0 时，每向 S0BUF 写入一个字符只产生一次发送中断。当双缓冲被禁止时，该位必须为 0。注：除了第一个写入的字符（S0BUF 为空时）之外，所有发送中断的位置都由 INTLO_0 决定。当写入第一个字符时，在 S0BUF 被写入后立即产生发送中断。
5	CIDIS_0	组合中断禁止 0。该位置位时使用各自独立的 Rx 和 Tx 中断。该位清零时 UART 选择 Tx/Rx 组合中断（与传统 80C51 UART 相同）。该位复位时为 0，选择组合中断。
6	INTLO_0	发送中断位置 0。0：在停止位的开始产生 Tx 中断；1：在停止位的结束产生 Tx 中断。模式 0 时必须为 0。注意：在使用单缓冲时，如果在停止位的结束产生 Tx 中断，在下一个起始位之前可能存在一个间隙。
7	DBMOD_0	双缓冲模式 0。该位置位时使能双缓冲。UART 模式 0 时必须为 0。为了和现有的 80C51 器件兼容，该位复位时为 0 禁止双缓冲。

表 56　串口 1 状态寄存器（S1STAT-地址 D4h）位分配

位	7	6	5	4	3	2	1	0
符号	DBMOD_1	INTLO_1	CIDIS_1	DBISEL_1	FE_1	BR_1	OE_1	STINT_1
复位	x	x	x	x	x	x	0	0

表 57　串口 1 状态寄存器（S1STAT-地址 D4h）位描述

位	符号	描述
0	STINT_1	状态中断使能 1。该位置位时，FE_1, BR_1 或 OE_1 可产生中断。这些中断（向量地址 008Bh）与 RI（CIDIS_1=1 时）或组合 TI/RI（CIDIS_1=0）共用中断向量。该位清零时，FE_1, BR_1，OE_1 不产生中断。（注：FE_1, BR_1 或 OE_1 位通常和 RI_1 一起产生，不管 STINT 的状态如何，RI_1 都将产生中断。）注：如果 EBRR(AUXR1.6)置位，BR_1 可导致间隔检测复位。
1	OE_1	当接收缓冲区仍然为满时接收到一个新的字符（在软件读出缓冲区中的前一个字符之前），也就是当 RI_1 仍然为置位状态时接收到新字节的位 8，那么溢出错误 1 标志 OE_1 将置位。该位通过软件清零。
2	BR_1	间隔检测 1 标志。当连续检测到 11 个位都为低电平时表明检测到一个间隔。该位通过软件清零。
3	FE_1	帧错误 1 标志。当接收器在数据帧结束时没有收到有效的停止位时，该标志置位。通过软件清零。
4	DBISEL_1	双缓冲发送中断选择 1。只在双缓冲使能时使用。该位在双缓冲使能时控制产生中断的次数。当该位置位时，每向 S1BUF 写入一个字符后产生一个发送中断，另外在最后一个字符发送的停止位（即缓冲区已空）开始（INTLO_1=0）或结束（INTLO_1=1）时再产生一次发送中断。最后这次中断可用于指示所有发送操作已完成。当该位为 0 时，每向 S1BUF 写入一个字符只产生一次发送中断。当双缓冲被禁止时，该位必须为 0。注：除了第一个写入的字符（S1BUF 为空时）之外，所有发送中断的位置都由 INTLO_1 决定。当写入第一个字符时，在 S1BUF 被写入后立即产生发送中断。
5	CIDIS_1	组合中断禁止 1。该位置位时使用各自独立的 Rx 和 Tx 中断。该位清零时 UART 选择 Tx/Rx 组合中断（与传统 80C51 UART 相同）。该位复位时为 0，选择组合中断。
6	INTLO_1	发送中断位置 1。0：在停止位的开始产生 Tx 中断；1：在停止位的结束产生 Tx 中断。模式 0 时必须为 0。注意：在使用单缓冲时，如果在停止位的结束产生 Tx 中断，在下一个起始位之前可能存在一个间隙。
7	DBMOD_1	双缓冲模式 1。该位置位时使能双缓冲。UART 模式 0 时必须为 0。为了和现有的 80C51 器件兼容，该位复位时为 0 禁止双缓冲。

10.10 更多关于 UART 模式 0 的信息

在模式 0 中，执行写 SnBUF 的指令将会启动发送。在发送结束时，TI_n(SnCON.1)置位，该位必须由软件清零。在该模式中必须禁止双缓冲。

清零 RI_n(SnCON.0)将启动接收。开始同步串行传送并且在传送结束时 RI_n 将再次置位。当 RI_n 清零时，开始下一个字符的接收，相关时序见图 21。

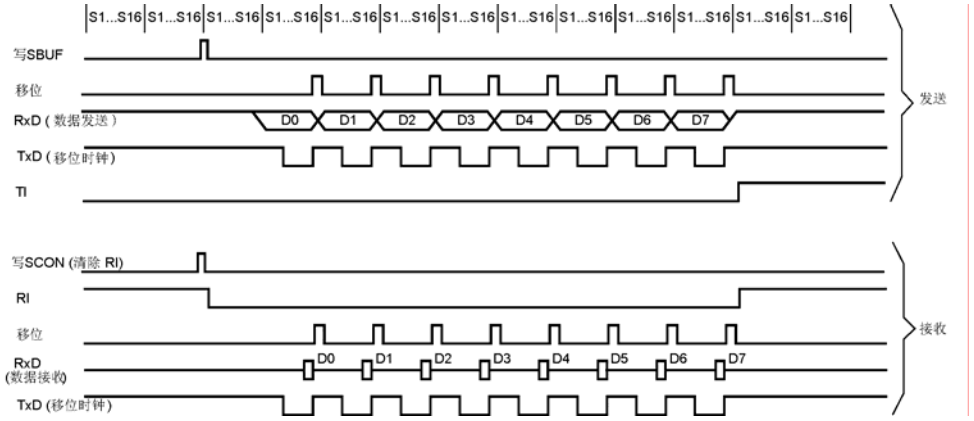


图 21　串口模式 0（双缓冲必须禁止）

10.11 更多关于 UART 模式 1 的信息

在模式 1 中，检测到 RxDn 端电平 1 到 0 跳变时启动接收操作。位检测器以波特率的 16 倍对 RxDn 进行采样。当检测到负跳变时，16 分频计数器立即复位。每个位时间被分成 16 个计数状态。在第 7、8、9 计数状态时，位检测器对 RxDn 端的值进行采样。三个采样值中至少有 2 个相同时，该值才能作为读入值，这样做是为了抑制噪声。如果在第一个位时间所接收的位不为 0，接收电路复位，接收器继续等待另一个负跳变的到来。这样可以防止错误的起始位。如果起始位被证明是有效的，则被移入输入移位寄存器，并开始接收这一帧剩余的位。

当且仅当产生最后一位移位脉冲时满足下列条件：RI_n=0 以及 SM2_n=0 或接收到的停止位=1，数据才会被装入 SnBUF 和 RB8_n，并置位 RI_n。上述两个条件中的任何一个不满足，所接收到的数据帧都会丢失。两个条件都满足时，停止位就进入 RB8_n，而 8 位数据则进入 SnBUF，并且 RI_n 被激活。

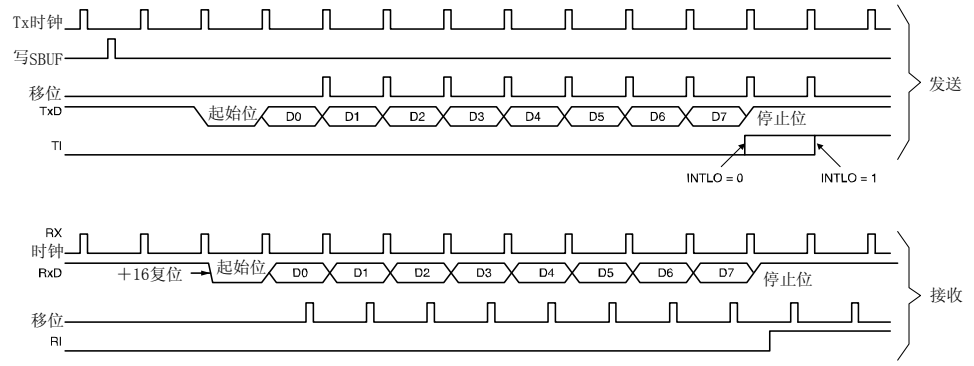


图 22　串口模式 1（单缓冲）

10.12 更多关于模式 2 和模式 3 的信息

接收的工作方式与模式 1 相同。

当且仅当产生最后一位移位脉冲时满足下列条件：(a) RI_n=0，以及(b) SM2_n=0 或接收到的第 9 位数据=1，数据才会被装入 SnBUF 和 RB8_n，并置位 RI_n。上述两个条件中的任何一个不满足，所接收到的数据帧都会丢失，RI_n 不被置位。两个条件都满足时，接收到的第 9 位数据就进入 RB8_n，而前 8 位数据则进入 SnBUF。

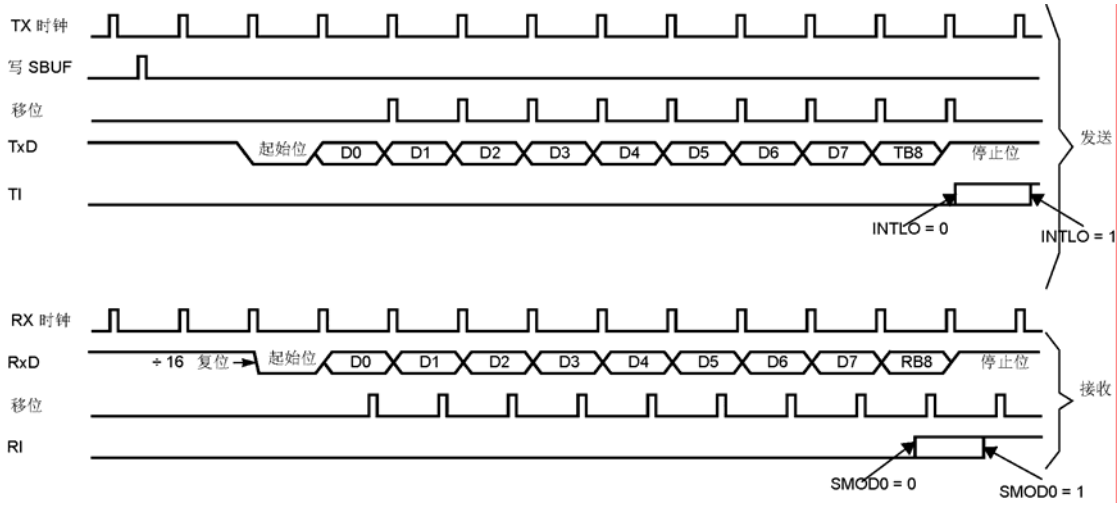


图 23　串口模式 2 或 3（单缓冲）

10.13 模式 2 和 3 中 SM2_n=1 时的帧错误和 RI_n

如果在模式 2 和 3 中 SM2_n=1，RI_n 和 FE_n 的操作如下表所示。

表 58　模式 2 和 3 中 SM2_n=1 时 FE_n 和 RI_n 的状态

模式	PCON.6 (SMOD0)	RB8_n	RI_n	FE_n
2	0	0	当 RB8_n=0 时无 RI_n	发生在停止位
		1	和图 23 相似，SMOD0=0，RI_n 在 RB8_n，FE_n 的前一位时产生	发生在停止位
3	1	0	当 RB8_n=0 时无 RI_n	不会发生
		1	和图 23 相似，SMOD0=1，RI_n 在停止位时发生	发生在停止位

10.14 间隔检测

当检测到连续 11 个位都为低时看成是检测到间隔，间隔检测在状态寄存器（SnSTAT）中报告。在模式 1 中，11 个字符由起始位、8 个数据位和 2 个停止位组成。在模式 2 和 3 中，11 个字符由起始位、9 个数据位和 1 个停止位组成。间隔检测位通过软件或复位清零。UART0 间隔检测可用来复位器件和使器件强制进入 ISP 模式，前提是 UART0 使能且 EBRR 位（AUXR1.6）置位和产生间隔。

10.15 双缓冲

UART 具有一个发送双缓冲器，这就允许在发送第一个字符的时候向 SBUF 写入第二个字符。双缓冲允许任意两个字符之间只有一个停止位的字符串的发送，只要下个字符在前一个字符的起始位和停止位之间写入 SnBUF。

双缓冲可以被禁止。当禁止时（DBMOD_n，即 SnSTAT.7=0），UART 和传统的 80C51 UART 兼容。如果使能该功能，UART 允许在前一个数据移位发送的过程中向 SnBUF 写入新数据。

10.16 不同模式中的双缓冲

只有在模式 1、2 和 3 中才可以使能双缓冲。当处于模式 0 时，必须禁止双缓冲（DBMOD_n=0）。

10.17 双缓冲使能时的发送中断（模式 1，2 和 3）

与传统的 UART 不同的是，当使能双缓冲时，Tx 中断发生在双缓冲器准备接收新数据的时候。发送的过程是这样的（假设为 8 个数据位）：

1. 双缓冲器初始化为空。
2. CPU 将数据写入 SnBUF
3. SnBUF 的数据装入移位寄存器并立即产生一个 Tx 中断。
4. 如果有后续的数据，执行第 6 步，否则执行第 5 步。
5. 如果没有后续的数据，那么：

   － 如果 DBISEL_n=0，不再产生任何中断。

   － 如果 DBISEL_n=1 且 INTLO_n=0，在移位寄存器中当前数据（也是最后的数据）停止位的开始产生 Tx 中断。

   － 如果 DBISEL_n=1 且 INTLO_n=1，在移位寄存器中当前数据（也是最后的数据）停止位的结束产生 Tx 中断。

   － 注意：在上一个数据的停止位移出时，如果 DBISEL_n＝1，而且 CPU 正将数据写入 SnBUF，这样就会出现一个不确定的状况。即，在 UART 并不确定是否还有更多的数据要传送的时候，Tx（发送）中断或许已经产生了。
6. 如果有后续的数据，CPU 再次将数据写入 SnBUF。那么：

   － 如果 INTLO_n=0，装入新数据并在移位寄存器中当前数据停止位的开始产生 Tx 中断。

   － 如果 INTLO_n=1，装入新数据并在移位寄存器中当前数据停止位的结束产生 Tx 中断。

   －跳到第 3 步。

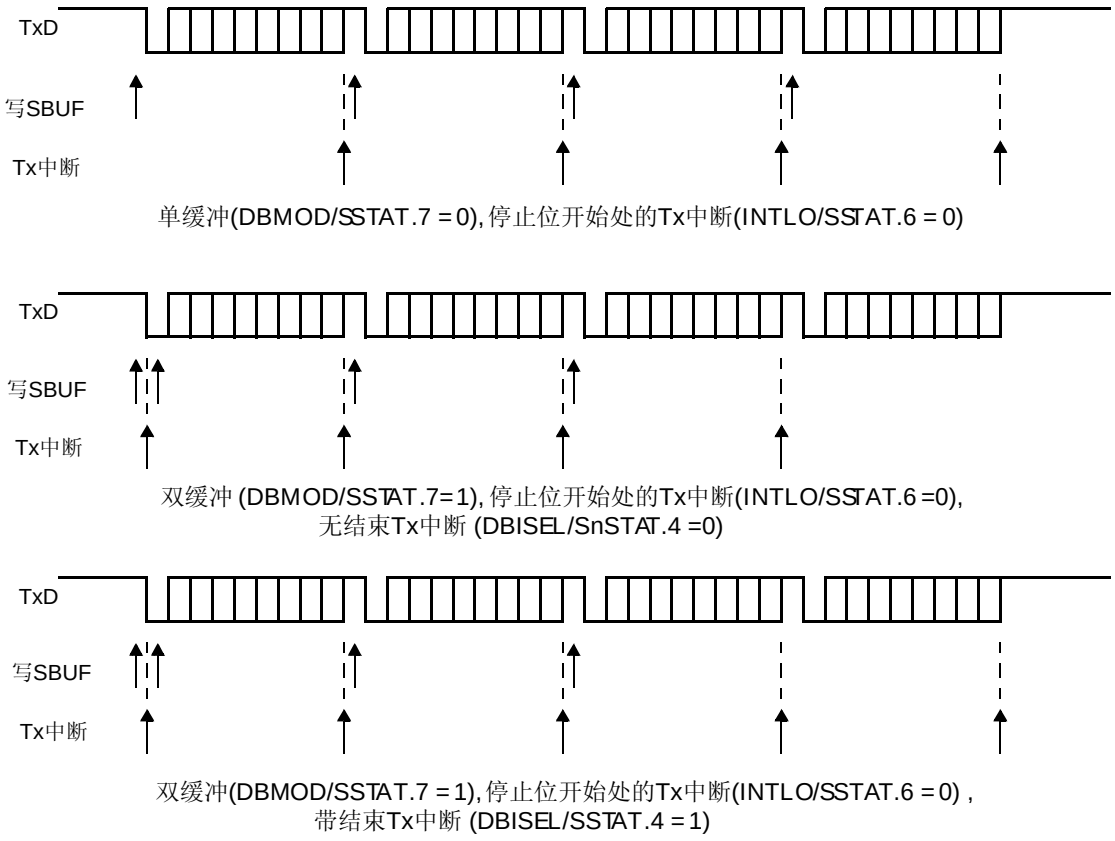


图 24　带双缓冲的发送和不带双缓冲的发送

10.18 双缓冲中的第 9 位数据（模式 1、2 和 3）

如果双缓冲被禁止(DBMOD_n，即 SnSTAT.7=0)，对 TB8_n 的写操作可以在写入 SnBUF 之前或之后进行，只要在 TB8_n 移出之前被更新即可。TB8_n 只能在移出之后（通过 Tx 中断指示）才能更改。

如果双缓冲使能，TB8_n 必须在写 SnBUF 之前更新，因为 TB8_n 将和 SnBUF 的数据一起双缓冲。10.17 节“具有双缓冲使能的发送中断（模式 1，2 和 3）”所描述的发送操作变成下面这样：

1. 双缓冲器初始化为空。
2. CPU 对 TB8_n 进行写操作。
3. CPU 将数据写入 SnBUF。
4. SnBUF/TB8_n 的数据装入移位寄存器并立即产生一个 Tx 中断。
5. 如果有后续的数据，执行第 7 步，否则执行第 6 步。
6. 如果没有后续的数据，那么：

— 如果 DBISEL_n=0，不再产生任何中断。

— 如果 DBISEL_n=1 且 INTLO_n=0，在移位寄存器中当前数据（也是最后的数据）发送停止位的开始产生 Tx 中断。

— 如果 DBISEL_n=1 且 INTLO_n=1，在移位寄存器中当前数据（也是最后的数据）发

送停止位的结束产生 Tx 中断。

7. 如果有后续的数据，CPU 再次写 TB8_n。

8. CPU 再次将数据写入 SnBUF。那么：

－ 如果 INTLO_n=0，装入新数据并在移位寄存器中当前数据停止位发送的开始产生 Tx 中断。

－ 如果 INTLO_n=1，装入新数据并在移位寄存器中当前数据停止位发送的结束产生 Tx 中断。

9. 跳到第 4 步。

10. 注：在上一个数据的停止位移出时，如果 DBISEL_n＝1，而且 CPU 正将数据写入 SnBUF，这样就会出现一个不确定的状况。即，在 UART 并不确定是否还有更多的数据要传送的时候，Tx（发送）中断或许已经产生了。

10.19 多机通信

UART 模式 2 及模式 3 有一个专门的应用领域，即多机通信。在这些模式下，发送及接收均为 9 位数据。接收时第 9 位数据存入 RB8_n。UART 可设置，以便在接收停止位的过程中仅当 RB8_n=1 时串口中断才激活。可通过置位 SnCON 内 SM2_n 位来使能这一特性。下面讲述多机系统使用该特性的一种方法：

当主机需要发送一数据块给某一台从机时，首先发送一个地址字节以识别目标从机。地址字节与数据字节的区别在于第 9 位数据，地址字节的第 9 位为 1，而数据字节的第 9 位为 0。SM2_n=1 时，数据字节不会使从机产生中断，而地址字节则会使所有从机都产生中断，这样每个从机可以检查接收到的字节并判断是否被寻址。被寻址的从机将清零 SM2_n 位以准备接收随后的数据字节。未被寻址的从机的 SM2_n 位仍为 1，这样就忽略随后的数据字节继续各自工作。

注：在模式 0 中 SM2_n 无效，在模式 1 中 SM2_n 必须为 0。

10.20 自动地址识别

自动地址识别是这样一种特性，它允许使用硬件进行比较，使 UART 从串行数据流中识别出特定的地址。这样就不必花费大量软件资源去检查每一个通过串口的串行地址。将 SnCON 内 SM2_n 置位可使能该特性。在 9 位 UART 模式（模式 2 和模式 3）下，如果接收的字节中包含“给定”地址或“广播”地址，接收中断标志（RI_n）将自动置位。在 9 位模式下要求第 9 个信息位为 1 以表明该信息内容是地址而非数据。

使用自动地址识别特性时，主机通过调用给定的从机地址选择与一个（或多个）从机通信。使用广播地址可以联系所有的从机。有两个特殊功能寄存器用来定义从机地址：SnADDR 和地址屏蔽 SnADEN。SnADEN 用于定义 SnADDR 内哪几位需要使用而哪几位是无关位。SnADEN 屏蔽可以与 SnADDR 逻辑“与”得出给定的地址，主机使用该地址对每一从机进行寻址。使用给定地址可以识别多个从机而排除另外的从机。下面的例子就显示了该方案的通用性：

表 59　从机 0/1 举例

例 1		例 2	
从机 0	SnADDR=1100 0000	从机 1	SnADDR=1100 0000
	SnADEN=1111 1101		SnADEN=1111 1110
	给定地址=1100 00x0		给定地址=1100 000x

上例中 SnADDR 相同，而 SnADEN 的数据用于区分两个从机。从机 0 要求位 0 为 0 而忽略位 1。从机 1 则要求位 1 为 0 而忽略位 0。由于从机 1 要求位 1 必须为 0，从机 0 唯一的地址应当是 1100 0010。由于要排除从机 0，位 0 必须为 1，因此从机 1 唯一的地址应当为 1100 0001。通过使地址的位 0 为 0（对于从机 0），位 1 为 0（对于从机 1），可同时选择从机 0 和从机 1。而取地址 1100 0000 时两从机都可被寻址。

下例所示为选择从机 1、2 而不选从机 0：

表 60　从机 0/1/2 举例

例 1		例 2		例 3	
从机 0	SnADDR=1100 0000	从机 1	SnADDR=1110 0000	从机 2	SADDR=1100 0000
	SnADEN=1111 1001		SnADEN=1111 1010		SADEN=1111 1100
	给定地址=1100 0xx0		给定地址=1110 0x0x		给定地址=1110 00xx

上述三个从地址只有低 3 位不同。从机 0 要求位 0=0，它可通过 1110 0110 单独寻址；从机 1 要求位 1 为 0，可通过 1110 0101 单独寻址；从机 2 要求位 2 为 0，可通过 1110 0011 单独寻址。由于必须使地址字节的第 2 位为“1”以屏蔽从机 2，因此使用地址 1110 0100 可选择从机 0 和 1 同时屏蔽从机 2。将 SnADDR 和 SnADEN 相“或”后产生每个从机的“广播”地址，此处结果为零的位视为无关位。大多数情况下，无关位被解释为 1，这样，“广播”地址为 FFH。复位时 SnADDR 和 SnADEN 均为 00H，此时产生了一个所有位都是无关位的给定地址，以及所有位都“无关”的广播地址。这样有效地禁止了自动寻址模式，允许微处理器使用不带有上述特性的标准 UART 驱动器。

11.I²C 接口

I²C 总线用两条线，串行时钟（SCL）和串行数据（SDA），在与总线相连的器件之间传递信息。总线的主要特性如下：

- 主机和从机之间为双向数据传送
- 多主机总线（无中央主机）
- 多主机同时传送时进行仲裁避免总线上数据冲突
- 串行时钟同步使得不同位速率的器件可以通过一条串行总线进行通信
- 串行时钟同步可作为握手机制，用于挂起和恢复串行传输
- I²C 总线可用于测试和诊断

典型的 I²C 总线配置如图 25 所示。根据方向位（R/W）状态的不同，I²C 总线上具有两种不同类型的数据传输：

- 数据从主发送器传送到从接收器。主机发送的第一个字节为从地址，接下来为数据字节。从机每接收一个字节就返回一个应答位。

- 数据从从发送器传送到主接收器。第一个字节（从地址）由主机发送，然后从机返回一个应答位。接下来是从机向主机发送的数据字节。主机每接收一个字节会返回一个应答位，但最后一个字节除外。接收完最后一个字节时返回一个“非应答”。主机产生所有的串行时钟脉冲以及起始和停止条件。传输以一个停止条件或一个重复的起始条件结束。由于重复的起始条件是下一次传输的开始，因此 I^2C 总线不会被释放。

P89LPC952 器件提供字节方式的 I^2C 接口。它支持 4 种工作模式：主发送模式、主接收模式、从发送模式和从接收模式。

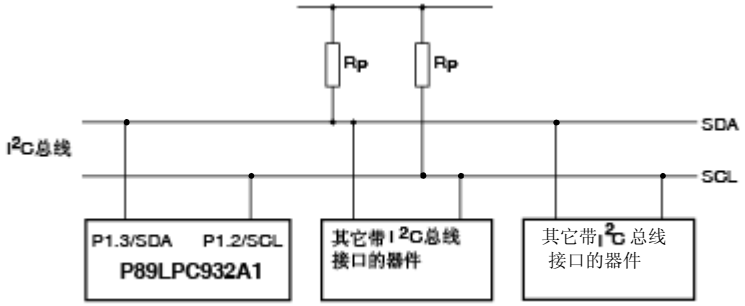


图 25　I^2C 总线配置

P89LPC952 CPU 通过下面 6 个特殊功能寄存器实现接口：I2CON（I^2C 控制寄存器）、I2DAT（I^2C 总线数据寄存器）、I2STAT（I^2C 总线状态寄存器）、I2ADR（I^2C 总线从地址寄存器）、I2SCLH（SCL 占空比寄存器高字节）和 I2SCLL（SCL 占空比寄存器低字节）。

11.1 I^2C 总线数据寄存器

I2DAT 寄存器包含了要发送的数据和刚接收到的数据。当 CPU 没有进行移位处理时，可以读这个 8 位寄存器。这就意味着用户只有在 SI 置位时才能对 I2DAT 进行访问。SI 位置位时 I2DAT 的数据一直保持不变。I2DAT 的数据总是从右向左移动：最先发送 MSB 位（位 7），接收完一个字节后，接收到的第一个数据位置于 I2DAT 的 MSB 位。

表 61　I^2C 数据寄存器（I2DAT-地址 DAh）位分配

位	7	6	5	4	3	2	1	0
符号	I2DAT.7	I2DAT.6	I2DAT.5	I2DAT.4	I2DAT.3	I2DAT.2	I2DAT.1	I2DAT.0
复位	0	0	0	0	0	0	0	0

11.2 I^2C 从地址寄存器

CPU 可对 I2ADR 寄存器进行读或写操作。该寄存器仅在 I^2C 接口处于从模式时才使用。主模式中该寄存器无效。I2ADR 的最低位（LSB）为通用调用位。当该位置位时对通用调用地址（00H）进行识别。

表 62　I²C 从机地址寄存器（I2ADR-地址 DBh）位分配

位	7	6	5	4	3	2	1	0
符号	I2ADR.6	I2ADR.5	I2ADR.4	I2ADR.3	I2ADR.2	I2ADR.1	I2ADR.0	GC
复位	0	0	0	0	0	0	0	0

表 63　I²C 从机地址寄存器（I2ADR-地址 DBh）位描述

位	符号	描述
0	GC	通用调用位。置位时，对通用调用地址进行识别，否则忽略。
1:7	I2ADR 1:7	7 位自身从地址。当处于主模式时，该寄存器的内容无效。

11.3 I²C 总线控制寄存器

CPU 可对该寄存器进行读或写操作。其中有 2 位受硬件影响：SI 位和 STO 位。SI 位由硬件置位，而 STO 位由硬件清零。

当 I²C 处于主模式时，CRSEL 决定 SCL 时钟源，在从模式中，该位被忽略并自动与主 I²C 器件的时钟频率（最大为 400kHz）同步。当 CRSEL＝1 时，I²C 接口将定时器 1 的溢出速率的 1/2 作为 I²C 时钟频率。定时器 1 应由用户设置为 8 位自动重装模式（模式 2）。

I²C 数据传输速率 ＝ 定时器溢出速率/2 ＝ PCLK/(2×(256－重装值))

如果 fosc＝12MHz，重装值为 0～255，则 I²C 数据传输速率范围为 11.72～3000Kbit/sec。

当 CRSEL=0 时，I²C 接口使用内部时钟发生器（其频率由 I2SCLL 和 I2SCLH 寄存器的值决定）作为时钟源。占空比并不一定要为 50%。

STA 位为起始标志。该位的置位将使 I²C 接口进入主模式，并在进入主模式后发送一个起始条件和重复起始条件。

STO 位为停止标志。在主模式中置位该位将使 I²C 接口发送一个停止条件，在从模式中置位该位使 I²C 接口从一个错误状态中恢复。

当 STA 和 STO 都置位时，如果是在主模式中，将向 I²C 总线发送一个停止条件，然后再发送一个起始条件；如果是在从模式中，将产生一个内部停止条件，但不发送到总线。

表 64　I²C 控制寄存器（I2CON-地址 D8h）位分配

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	CRSEL
复位	x	0	0	0	0	0	x	0

表 65 I^2C 控制寄存器（I2CON-地址 D8h）位描述

位	符号	描述
0	CRSEL	SCL 时钟选择。CRSEL=1：定时器 1 溢出产生 SCL；CRSEL＝0：使用内部 SCL 发生器（其频率由 I2SCLH 和 I2SCLL 的值决定）。
1	-	保留。
2	AA	发送应答标志。当该位置位时，下面的任意条件下，在 SCL 线的应答时钟脉冲内将返回一个应答（SDA 上的低电平）： (1)收到自身的从地址。(2)在 I2ADR 的通用调用位（GC）置位时接收到通用调用地址。(3)当 I^2C 接口处于主接收器模式时，接收到一个数据字节。(4)当 I^2C 接口处于可寻址的从接收器模式时，接收到一个数据字节。 当该位清零时，下面的任意条件下，在 SCL 线的应答时钟脉冲内将返回一个非应答标志（SDA 上的高电平）： (1)当 I^2C 接口处于主接收器模式时，接收到一个数据字节。(2)当 I^2C 接口处于可寻址的从接收器模式时，接收到一个数据字节。
3	SI	I^2C 中断标志。当进入 25 种可能的 I^2C 状态中的其中一个后，该位置位。当 EA, EI2C(IEN1.0)以及 SI 置位时，产生一个中断请求。SI 必须通过软件清零。
4	STO	停止标志。STO=1：在主模式中，向 I^2C 总线发送一个停止条件。I^2C 总线检测到停止条件时将自动清零该位。在从模式中，该位的置位可使总线从错误状态中恢复。该情况下，不向总线发送停止条件。硬件的操作就好像是接收到一个停止条件并切换到“不可寻址”的从接收器模式。STO 标志由硬件自动清零。
5	STA	起始标志。STA=1：I^2C 进入主模式，检测 I^2C 总线并当总线空闲时产生一个起始条件。如果总线忙，则等待一个停止条件（该条件能释放总线）并在延迟半个内部时钟发生器周期后发送一个起始条件。当 I^2C 接口已经处于主模式中并发送或接收了一些数据时，该位置位会发送一个重复的起始条件。STA 可在任何时候置位，当 I^2C 接口处于可被寻址的从模式时，STA 也可以置位。STA=0：不会产生起始或重复起始条件。
6	I2EN	I^2C 接口使能。该位置位时，使能 I^2C 接口；该位为 0 时，I^2C 功能被禁止。
7	-	保留。

11.4 I^2C 状态寄存器

I2STAT 是一个只读寄存器。它包含了 I^2C 接口的状态代码。最低 3 位始终为 0。I^2C 一共有 26 种可能的状态代码。当代码为 F8H 时表示无可用的相关状态信息，SI 也不会置位。所有其它 25 种状态码都对应一个已定义的 I^2C 状态。进入任何一个状态时，SI 都会置位。详见表 71 到表 74。

表 66 I^2C 状态寄存器（I2STAT-地址 D9h）位分配

位	7	6	5	4	3	2	1	0
符号	STA.4	STA.3	STA.2	STA.1	STA.0	0	0	0
复位	0	0	0	0	0	0	0	0

表 67 I^2C 状态寄存器（I2STAT-地址 D9h）位描述

位	符号	描述
0:2	-	保留，总是设为 0。
3:7	STA[0:4]	I^2C 总线状态代码。

11.5 I^2C SCL 占空比寄存器 I2SCLH 和 I2SCLL

当设置 I2CON 寄存器的 CRSEL＝0 来选择内部 SCL 发生器作为 I^2C 接口的时钟源时，用户必须对 I2SCLL 和 I2SCLH 进行设置以选择合适的数据传输速率。I2SCLH 定义 SCL 高电平的 PCLK 周期数，I2SCLL 定义 SCL 低电平的 PCLK 周期数。频率由下面的公式决定：

位速率＝f_{PCLK}/(2×(I2SCLH + I2SCLL))

此处，f_{PCLK} 表示 PCLK 的频率。

I2SCLL 和 I2SCLH 的值不一定要相同；用户可通过设置这两个寄存器得到不同的 SCL 占空比。但寄存器的值必须确保所得的数据率在 I^2C 的 0～400kHz 的数据率范围内。因此 I2SCLL 和 I2SCLH 的值有一些限制，建议两个寄存器的取值都应大于 3 个 PCLK 周期。

表 68 I^2C 时钟速率选择

I2SCLL+ I2SCLH	CRSEL	fosc 取不同值时的 I^2C 位数据率（Kbit/s）				
		7.373MHz	3.6865MHz	1.8433MHz	12MHz	6MHz
6	0	-	307	154	-	-
7	0	-	263	132	-	-
8	0	-	230	115	-	375
9	0	-	205	102	-	333
10	0	369	184	92	-	300
15	0	246	123	61	400	200
25	0	147	74	37	240	120
30	0	123	61	31	200	100
50	0	74	37	18	120	60
60	0	61	31	15	100	50
100	0	37	18	9	60	30
150	0	25	12	6	40	20
200	0	18	9	5	30	15
-	1	3.6-922Kbps 模式 2 中的定时器 1	1.8-461Kbps 模式 2 中的定时器 1	0.9-230Kbps 模式 2 中的定时器 1	5.86-1500Kbps 模式 2 中的定时器 1	2.93-750Kbps 模式 2 中的定时器 1

11.6 I²C 操作模式

11.6.1 主发送器模式

在该模式中，数据从主机发送到从机。在进入主发送器模式之前，I2CON 必须按照下表进行初始化：

表 69 I²C 控制寄存器（I2CON-地址 D8h）

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	CRSEL
值	-	1	0	0	0	x	-	位速率

CRSEL 定义串行位速率。I2EN 必须置位以使能 I²C 功能。AA 位为 0 时，如果另一个器件成为总线的主机，I²C 将不会对它自身的从地址或通用调用地址产生应答。它不能进入从模式。STA，STO 和 SI 必须设置为 0。

第一个发送的数据包含接收器件的从地址（7 位）和数据方向位。在此模式下，数据方向位（R/W）应当为 0 表示执行写操作。每次发送 8 位数据。每发送完一个字节，都接收到一个应答位。输出起始和停止条件用于指示串行传输的开始和结束。

I²C 总线通过置位 STA 将进入主发送器模式。在总线空闲后 I²C 逻辑立即发送一个起始条件。当发送完起始条件后，SI 置位。此时状态寄存器（I2STAT）中的状态代码应当为 08H。该状态代码用于指向一个中断服务程序。在该中断程序中，用户将从地址和数据方向位（SLA+W）装入 I2DAT（数据寄存器））。SI 位必须在数据传输能够继续进行之前清零。

当从地址和 R/W 位已发送且接收到应答位之后，SI 位再次置位，并且对于主模式，可能的状态代码为 18H，20H 或 38H，如果使能从模式（AA=1），可能的状态代码为 68H，78H 或 0B0H。每个状态代码对应的执行动作见表 83。

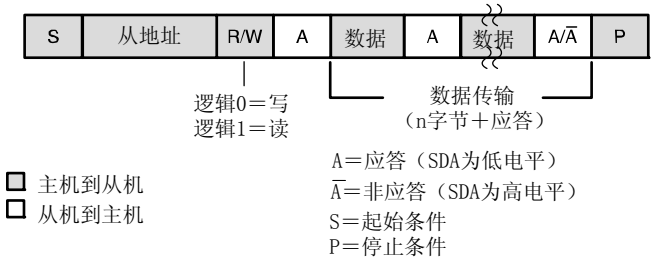


图 26 主发送器模式中的格式

11.6.2 主接收器模式

在主接收器模式中，数据字节的接收来自从发送器。传输的初始化与主发送器模式相同。发送完起始条件后，中断服务程序必须将 7 位从地址和数据方向位装入 I²C 数据寄存器（I2DAT）。SI 位必须在数据传输能够继续进行之前清零。

当从地址和方向位已发送且接收到应答位之后，SI 置位而状态寄存器将显示状态代码。对于主模式，可能的状态代码为 40H，48H 或 38H，对于从模式，可能的状态代码为 68H，78H 或 B0H。详见表 73。

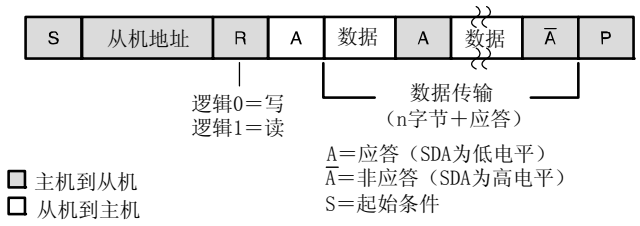


图 27　主接收器模式的格式

在一个重复的起始条件之后，I^2C 可以切换到主发送器模式。

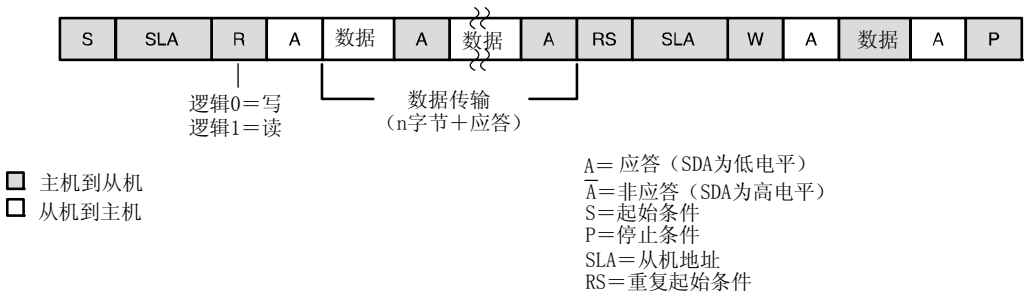


图 28　在发送重复起始条件后，主接收器切换到主发送器

11.6.3 从接收器模式

在从接收器模式中，从主发送器接收数据字节。要初始化从接收器模式，用户必须将从地址写入从地址寄存器（I2ADR）并按照下表配置 I^2C 控制寄存器（I2CON）：

表 70　I^2C 控制寄存器（I2CON-地址 D8h）

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	CRSEL
复位	-	1	0	0	0	1	-	-

CRSEL 不用于从模式。I2EN 必须置位以使能 I^2C 功能。AA 位必须置位以使 I^2C 应答自身的从地址或通用调用地址。STA，STO 和 SI 设置为 0。

当 I2ADR 和 I2CON 完成初始化时，I^2C 等待直到它被自身的从地址或通用地址寻址随后接收到数据方向位 0(W)。如果数据方向位为 1(R)，I^2C 将进入从发送器模式。在接收到自身从地址和方向位后，SI 置位并可从状态寄存器（I2STAT）中读出有效的状态代码。状态代码及其对应的执行动作见表 74。

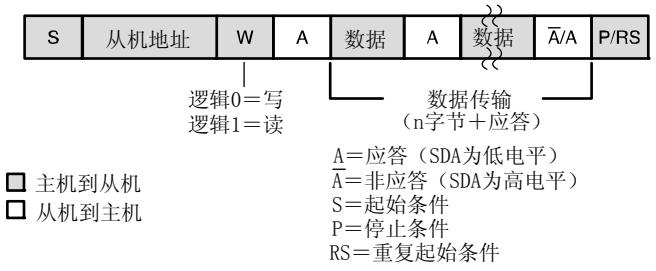


图 29　从接收器模式的格式

11.6.4 从发送器模式

第一个字节的接收和处理与从接收器模式相同。但在该模式中，方向位指示传输的方向掉转。串行数据通过 P1.3/SDA 发送而串行时钟通过 P1.2/SCL 输入。起始和停止条件被看成是传输的开始和结束。在一个给定的应用中，I^2C 可以为主模式也可以为从模式。在从模式中，I^2C 总线硬件寻找它自身的从地址和通用调用地址。如果检测到其中一个地址，将产生中断请求。当微控制器希望成为总线主机时，在进入主模式前硬件一直等待，直到总线释放。这样就不会中断一个可能的从机动作。如果在主模式中总线仲裁丢失，I^2C 将立即切换到从模式并能在同一个串行传输中检测自身的从地址。

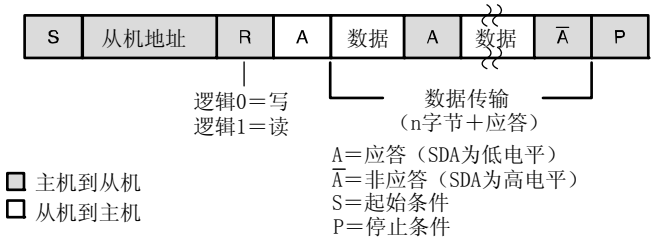


图 30 从发送器模式的格式

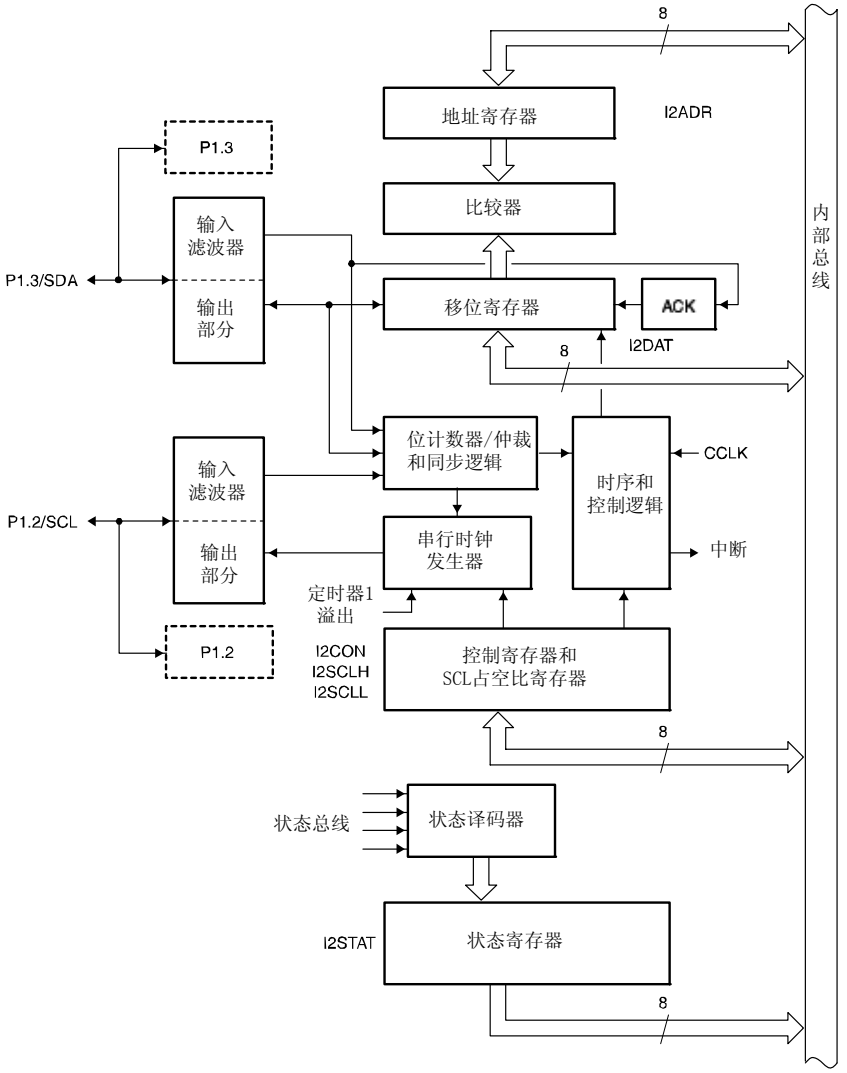


图 31 I^2C 总线串行接口框图

表 71　主发送器模式

状态代码(I2STAT)	I^2C 总线硬件状态	应用软件的响应					I^2C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
08H	已发送起始条件	装入 SLA+W	x	0	0	x	将发送 SLA+W，接收 ACK 位
10H	已发送重复起始条件	装入 SLA+W 或装入 SLA+R	x	0	0	x	同上；将发送 SLA+W；I^2C 将切换到主接收器模式
18H	已发送 SLA+W；已接收 ACK	装入数据字节或	0	0	0	x	将发送数据字节，接收 ACK 位
		无 I2DAT 动作或	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作或	0	1	0	x	将发送停止条件；STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件，然后发送起始条件；STO 标志将复位
20H	已发送 SLA+W；已接收非 ACK	装入数据字节或	0	0	0	x	将发送数据字节，接收 ACK 位
		无 I2DAT 动作或	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作或	0	1	0	x	将发送停止条件；STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件，然后发送起始条件；STO 标志将复位
28H	已发送 I2DAT 中的数据字节；已接收 ACK	装入数据字节或	0	0	0	x	将发送数据字节，接收 ACK 位
		无 I2DAT 动作或	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作或	0	1	0	x	将发送停止条件；STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件，然后发送起始条件；STO 标志将复位
30H	已发送 I2DAT 中的数据字节；已接收非 ACK	装入数据字节或	0	0	0	x	将发送数据字节，接收 ACK 位
		无 I2DAT 动作或	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作或	0	1	0	x	将发送停止条件；STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件，然后发送起始条件；STO 标志将复位
38H	在 SLA+R/W 或数据字节中丢失仲裁	无 I2DAT 动作或	0	0	0	x	I^2C 总线将被释放；进入不可寻址从模式
		无 I2DAT 动作	1	0	0	x	当总线变为空闲时发送起始条件

表 72　主接收器模式

状态代码 (I2STAT)	I^2C 总线硬件状态	应用软件的响应					I^2C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
08H	已发送起始条件	装入 SLA+R	x	0	0	x	将发送 SLA+R，接收 ACK 位
10H	已发送重复起始条件	装入 SLA+R 或	x	0	0	x	同上
		装入 SLA+W					将发送 SLA+W，I^2C 将切换到主发送器模式
38H	在非 ACK 位中丢失仲裁	无 I2DAT 动作	0	0	0	x	I^2C 总线将被释放；I^2C 将进入从模式
		无 I2DAT 动作	1	0	0	x	当总线恢复空闲后发送起始条件
40H	已发送 SLA+R；已接收 ACK	无 I2DAT 动作	0	0	0	0	将接收数据字节；返回非 ACK 位
		无 I2DAT 动作	0	0	0	1	将接收数据字节；返回 ACK 位
48H	已发送 SLA+R；已接收非 ACK	无 I2DAT 动作	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作	0	1	0	x	将发送停止条件；STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件，然后发送起始条件；STO 标志将复位
50H	已接收数据字节；已返回 ACK	读数据字节	0	0	0	0	将接收数据字节，返回非 ACK 位
		读数据字节	0	0	0	1	将接收数据字节；返回 ACK 位
58H	已接收数据字节；已返回 NACK	读数据字节	1	0	0	x	将发送重复起始条件
		读数据字节	0	1	0	x	将发送停止条件；STO 标志将复位
		读数据字节	1	1	0	x	将发送停止条件，然后发送起始条件；STO 标志将复位

表 73 从接收器模式

状态代码 (I2STAT)	I^2C 总线硬件状态	应用软件的响应					I^2C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
60H	已接收自身 SLA+W; 已接收 ACK	无 I2DAT 动作	x	0	0	0	将接收数据字节并返回非 ACK 位
		无 I2DAT 动作	x	0	0	1	将接收数据字节并返回 ACK 位
68H	主控器时在 SLA+R/W 中丢失仲裁;已接收自身 SLA+W,已返回 ACK	无 I2DAT 动作	x	0	0	0	将接收数据字节并返回非 ACK 位
		无 I2DAT 动作	x	0	0	1	将接收数据字节并返回 ACK 位
70H	已接收通用调用地址 (00H); 已返回 ACK	无 I2DAT 动作	x	0	0	0	将接收数据字节并返回非 ACK 位
		无 I2DAT 动作	x	0	0	1	将接收数据字节并返回 ACK 位

续表 73

状态代码 (I2STAT)	I^2C 总线硬件状态	应用软件的响应					I^2C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
78H	主控器时在 SLA+R/W 中丢失仲裁；已接收通用调用地址;已返回 ACK	无 I2DAT 动作	x	0	0	0	将接收数据字节并返回非 ACK 位
		无 I2DAT 动作	x	0	0	1	将接收数据字节并返回 ACK 位
80H	前一次寻址使用自身从地址；已接收数据字节；已返回 ACK	读数据字节	x	0	0	0	将接收数据字节并返回非 ACK 位
		读数据字节	x	0	0	1	将接收数据字节并返回 ACK 位
88H	前一次寻址使用自身 SLA 地址；已接收数据字节；已返回 ACK	读数据字节	0	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址
		读数据字节	0	0	0	1	切换到不可寻址 SLA 模式；识别自身 SLA；如果 I2ADR.0=1，将识别通用调用地址
		读数据字节	1	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		读数据字节	1	0	0	1	切换到不可寻址 SLA 模式；识别自身 SLA；如果 I2ADR.0=1，将识别通用调用地址；当总线空闲后发送起始条件
90H	前一次寻址使用通用调用；已接收数据字节；已返回 ACK	读数据字节	x	0	0	0	将接收数据字节并返回非 ACK 位
		读数据字节	x	0	0	1	将接收数据字节并返回 ACK 位
98H	前一次寻址使用通用调用；已接收数据字节；已返回非 ACK	读数据字节	0	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址
		读数据字节或	0	0	0	1	切换到不可寻址 SLA 模式；识别自身从地址；如果 I2ADR.0=1，将识别通用调用地址
		读数据字节或	1	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		读数据字节	1	0	0	1	切换到不可寻址 SLA 模式；识别自身从地址；如果 I2ADR.0=1，将识别通用调用地址;当总线空闲后发送起始条件

续表 73

状态代码 (I2STAT)	I^2C 总线硬件状态	应用软件的响应					I^2C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
A0H	当使用 SLA/REC 或 SLA/TRX 静态寻址时,接收到停止条件或重复的起始条件	无 I2DAT 动作	0	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址
		无 I2DAT 动作	0	0	0	1	切换到不可寻址 SLA 模式；识别自身从地址；如果 I2ADR.0=1，将识别通用调用地址
		无 I2DAT 动作	1	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 I2DAT 动作	1	0	0	1	切换到不可寻址 SLA 模式；识别自身从地址；如果 I2ADR.0═1，将识别通用调用地址；当总线空闲后发送起始条件

表 74　从发送器模式

状态代码 (I2STAT)	I^2C 总线硬件状态	应用软件的响应					I^2C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
A8H	已接收自身 SLA+R；已返回 ACK	装入数据字节或	x	0	0	0	将发送最后的数据字节并接收 ACK 位
		装入数据字节	x	0	0	1	将发送数据字节并接收 ACK 位
B0H	主控器时在 SLA+R/W 中丢失仲裁；已接收自身 SLA+R，已返回 ACK	装入数据字节或	x	0	0	0	将发送最后的数据字节并接收 ACK 位
		装入数据字节	x	0	0	1	将发送数据字节并接收 ACK 位
B8H	已发送 I2DAT 中数据字节；已接收 ACK	装入数据字节或	x	0	0	0	将发送最后的数据字节并接收 ACK 位
		装入数据字节	x	0	0	1	将发送数据字节并接收 ACK 位
C0H	已发送 I2DAT 中数据字节；已接收非 ACK	无 I2DAT 动作或	0	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址
		无 I2DAT 动作或	0	0	0	1	切换到不可寻址 SLA 模式；识别自身从地址；如果 I2ADR.0=1，将识别通用调用地址
		无 I2DAT 动作或	1	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 I2DAT 动作	1	0	0	1	切换到不可寻址 SLA 模式；识别自身从地址；如果 I2ADR.0=1，将识别通用调用地址；当总线空闲后发送起始条件

续表 74

状态代码 (I2STAT)	I²C 总线硬件状态	应用软件的响应					I²C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
C8H	已发送 I2DAT 中最后的数据字节（AA=0）；已接收 ACK	无 I2DAT 动作或	0	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址
		无 I2DAT 动作或	0	0	0	1	切换到不可寻址 SLA 模式；识别自身从地址；如果 I2ADR.0=1，将识别通用调用地址
		无 I2DAT 动作或	1	0	0	0	切换到不可寻址 SLA 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 I2DAT 动作	1	0	0	1	切换到不可寻址 SLA 模式；识别自身从地址；如果 I2ADR.0=1，将识别通用调用地址；当总线空闲后发送起始条件

12.串行外围接口（SPI）

P89LPC952 提供另一种高速串行通信接口——SPI 接口。SPI 是一种全双工、高速、同步的通信总线，有两种操作模式：主模式和从模式。在主模式或从模式中均支持高达 3Mbit/s 的速率。还具有传输完成标志和写冲突标志保护。

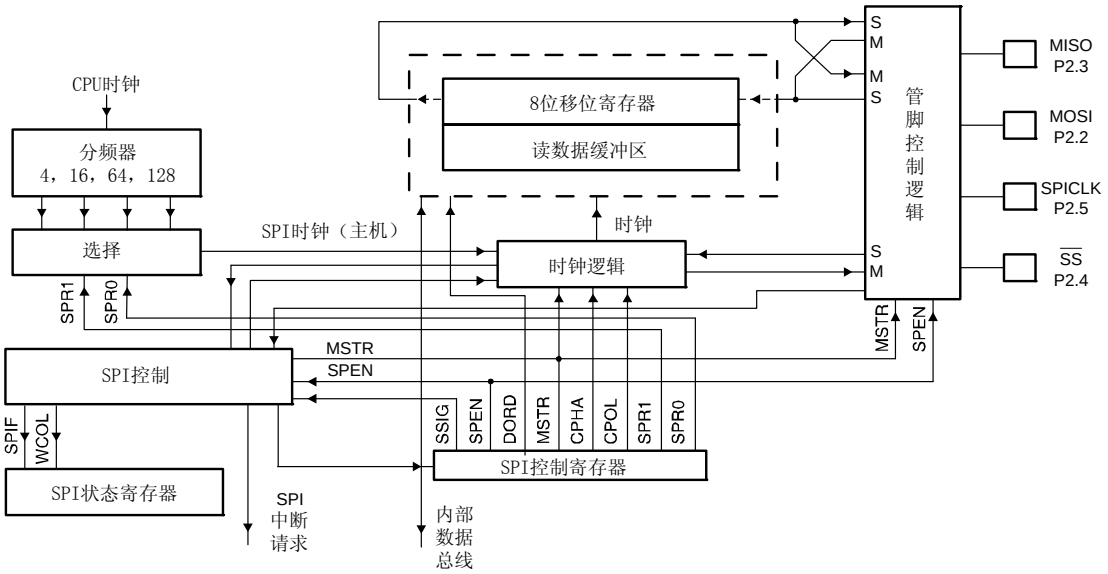


图 32　SPI 框图

SPI 接口有 4 个管脚：SPICLK, MOSI, MISO 和 $\overline{SS}$：

- SPICLK, MOSI 和 MISO 在两个或更多 SPI 器件中一般都是连接在一起的。数据通过 MOSI 从主机传送到从机（主机输出，从机输入），通过 MISO 由从机传送到主机（主机输入，从机输出）。SPICLK 信号在主模式时为输出，在从模式时为输入。如果 SPI 系统被禁止，即 SPEN(SPCTL.6)=0(复位值)，这些管脚都可作为 I/O 口使用。

- $\overline{SS}$ 为从机选择管脚。在典型的配置中，SPI 主机使用 I/O 口选择一个 SPI 器件作为当前的从机。SPI 从器件通过其 $\overline{SS}$ 脚确定是否被选择。如果满足下面的条件之一，$\overline{SS}$ 就被忽略：

  －如果 SPI 系统被禁止，即 SPEN(SPCTL.6)=0(复位值)

  －如果 SPI 配置为主机，即 MSTR(SPCTL.4)=1，并且 P2.4 配置为输出（通过 P2M1.4 和 P2M2.4）

  －如果 $\overline{SS}$ 脚被忽略，即 SSIG(SPCTL.7)位＝1，该脚配置用于 I/O 口功能。

注：即使 SPI 被配置为主机（MSTR＝1），它仍然可以通过拉低 $\overline{SS}$ 脚配置为从机（如果 P2.4 配置为输入且 SSIG=0）。要使能该特性，应当置位 SPIF(SPSTAT.7)（见 12.4 节 “$\overline{SS}$ 的模式改变）。

典型连接如图 33～35 所示。

表 75　SPI 控制寄存器位分配（SPCTL – 地址：E2h）

位	7	6	5	4	3	2	1	0
符号	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
复位	0	0	0	0	0	1	0	0

表 76　SPI 控制寄存器位描述（SPCTL – 地址：E2h）

位	符号	描述
0	SPR0	SPI 时钟速率选择。
1	SPR1	SPR1，SPR0： 00－CCLK/4 01－CCLK/16 10－CCLK/64 11－CCLK/128
2	CPHA	SPI 时钟相位选择（见图 42～图 45）： 1：数据在 SPICLK 的前时钟沿驱动，并在后时钟沿采样。 0：数据在 $\overline{SS}$ 为低（SSIG＝00）时被驱动，在 SPICLK 的后时钟沿被改变，并在前时钟沿被采样。（注：SSIG=1 时的操作未定义）
3	CPOL	SPI 时钟极性（见图 42～图 45）： 1－SPICLK 空闲时为高电平。SPICLK 的前时钟沿为下降沿而后沿为上升沿。 0－SPICLK 空闲时为低电平。SPICLK 的前时钟沿为上升沿而后沿为下降沿。
4	MSTR	主/从模式选择（见表 92）。
5	DORD	SPI 数据顺序： 1－数据字的 LSB 最先发送。 0－数据字的 MSB 最先发送。
6	SPEN	SPI 使能。 1：SPI 使能。 0：SPI 被禁止，所有 SPI 管脚都作为 I/O 口脚。
7	SSIG	$\overline{SS}$ 忽略。 1：MSTR（位 4）确定器件为主机还是从机。 0：$\overline{SS}$ 脚用于确定器件为主机还是从机。$\overline{SS}$ 脚可作为 I/O 口脚（见表 92）。

表 77　SPI 状态寄存器位分配（SPSTAT – 地址：E1h）

位	7	6	5	4	3	2	1	0
符号	SPIF	WCOL	-	-	-	-	-	-
复位	0	0	X	X	X	X	X	X

表 78　SPI 状态寄存器位描述（SPSTAT – 地址：E1h）

位	符号	描述
0:5	-	保留
6	WCOL	SPI 写冲突标志。在数据传输的过程中如果对 SPI 数据寄存器 SPDAT 执行写操作，WCOL 将置位（见 12.5 节“写冲突”）。WCOL 标志通过软件向其写入“1”清零。
7	SPIF	SPI 传输完成标志。当一次串行传输完成时，SPIF 置位，并当 ESPI(IEN1.3) 和 EA 都置位时产生中断。当 SPI 处于主模式且 SSIG=0 时，如果 $\overline{SS}$ 为输入并被驱动为低电平，SPIF 也将置位（见 12.4 节“通过 $\overline{SS}$ 改变模式”）。 SPIF 标志通过软件向其写入“1”清零。

表 79　SPI 数据寄存器的位分配（SPDAT – 地址：E3h）

位	7	6	5	4	3	2	1	0
符号	MSB	-	-	-	-	-	-	LSB
复位	0	0	0	0	0	0	0	0

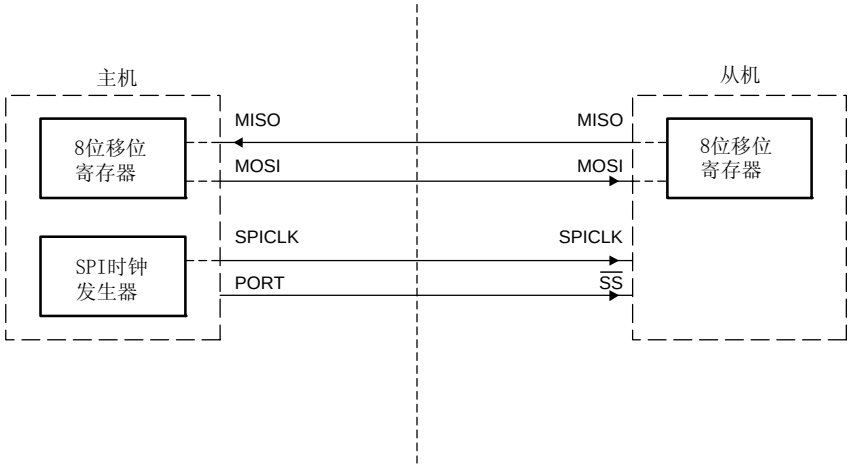


图 33　SPI 单主机单从机配置

在图 33 中，从机的 SSIG(SPCTL.7)为 0，$\overline{SS}$ 用于选择从机。SPI 主机可使用任何端口（包括 P2.4/$\overline{SS}$）来驱动 $\overline{SS}$ 脚。

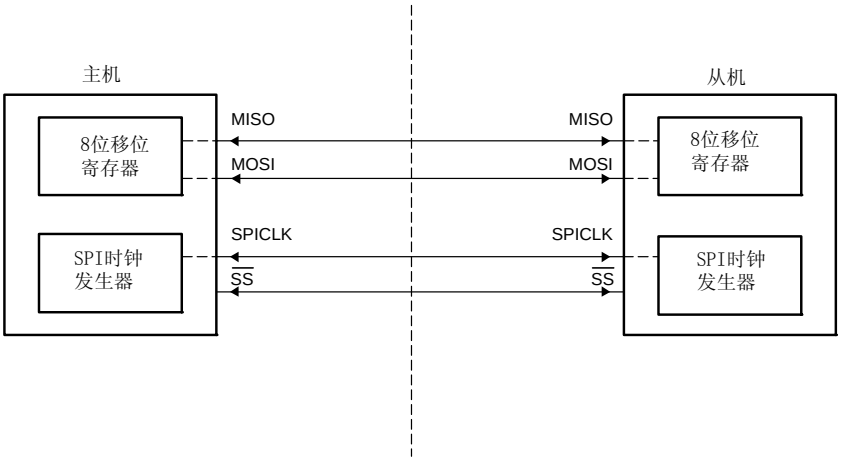


图 34　SPI 双器件配置（两器件可互为主从）

图 34 所示为两个器件互相连接且互为主从的情况。当没有发生 SPI 操作时，两个器件都可配置为主机（MSTR=1），将 SSIG 清零并将 P2.4($\overline{SS}$)配置为准双向模式。当其中一个器件启动传输时，它可将 P2.4 配置为输出并驱动为低电平，这样就强制另一个器件变为从机（见 12. 4 节“通过$\overline{SS}$改变模式”）。

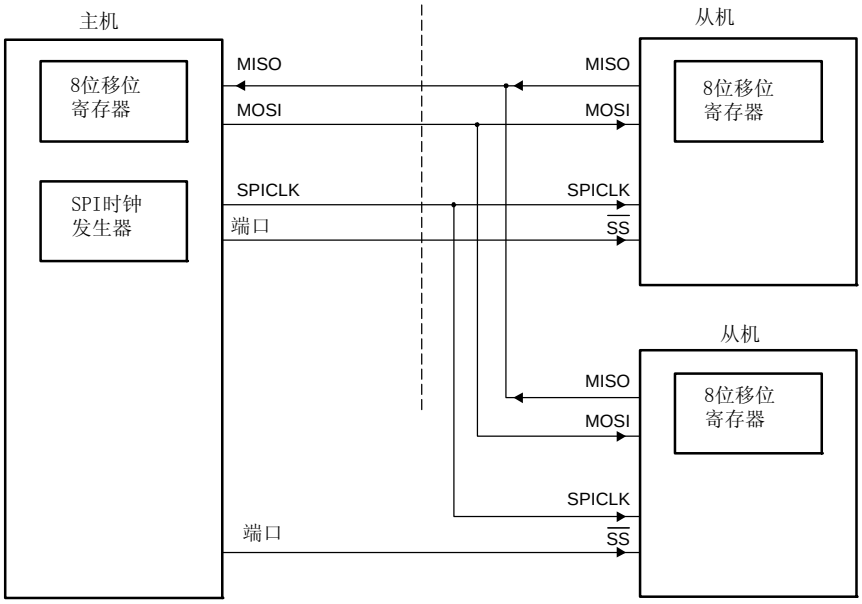


图 35　SPI 单主机多从机配置

在图 35 中，从机的 SSIG(SPCTL.7)为 0，从机通过对应的$\overline{SS}$信号被选中。SPI 主机可使用任何端口（包括 P2.4/$\overline{SS}$）来驱动$\overline{SS}$脚。

12.1 配置 SPI

表 80 所示为主/从模式的配置以及模式的使用和传输方向。

表 80　SPI 主从选择

SPEN	SSIG	$\overline{SS}$ 脚	MSTR	主或从模式	MISO	MOSI	SPICLK	备注
0	X	P2.4[1]	X	SPI 禁止	P2.3[1]	P2.2[1]	P2.5[1]	SPI 禁止。P2.2,P2.3,P2.4.P2.5 作为 I/O 口使用
1	0	0	0	从	输出	输入	输入	选择作为从机
1	0	1	0	从	高阻	输入	输入	未被选中。MISO 为高阻态以避免争夺总线
1	0	0	1(->0)[2]	从	输出	输入	输入	P2.4/ $\overline{SS}$ 配置为输入或准双向口。SSIG 为 0。如果选择 $\overline{SS}$ 并驱动为低电平，则被选择作为从机。当 $\overline{SS}$ 变为低电平时，MSTR 将清零。
1	0	1	1	主(空闲)	输入	高阻	高阻	当主机空闲时 MOSI 和 SPICLK 为高阻态以避免争夺总线。用户必须将 SPICLK 上拉或下拉（根据 CPOL-SPCTL.3 的取值）以避免出现悬浮状态。
				主(激活)		输出	输出	当主机被激活时，MOSI 和 SPICLK 为推挽输出
1	1	P2.4[1]	0	从	输出	输入	输入	
1	1	P2.4[1]	1	主	输入	输出	输出	

注：1.　作为 I/O 口使用

2.　当 $\overline{SS}$ 处于输入模式时被驱动为低电平且 SSIG=0 时，MSTR 位自动清零。

12.2 作为从机时的额外注意事项

当 CPHA＝0 时，SSIG 必须为 0，$\overline{SS}$ 脚必须取反并且在每个连续的串行字节之间重新设置为高电平。如果 SPDAT 寄存器在 $\overline{SS}$ 有效（低电平）时执行写操作，那么将导致一个写冲突错误。CPHA=0 且 SSIG=1 时的操作未定义。

当 CPHA＝1 时，SSIG 可以置位。如果 SSIG＝0，$\overline{SS}$ 脚可在连续传输之间保持低有效（即一直固定为低电平）。这种方式有时适用于具有单固定主机和单从机驱动 MISO 数据线的系统。

12.3 作为主机时的额外注意事项

在 SPI 中，传输总是由主机启动的。如果 SPI 使能（SPEN=1）并选择作为主机，主机对 SPI 数据寄存器的写操作将启动 SPI 时钟发生器和数据的传输。在数据写入 SPDAT 之后的半个到一个 SPI 位时间后，数据将出现在 MOSI 脚。

需要注意的是，主机可以通过将对应器件的 $\overline{SS}$ 脚驱动为低电平实现与之通信。写入主

机 SPDAT 寄存器的数据从 MOSI 脚移出发送到从机的 MOSI 脚。同时从机 SPDAT 寄存器的数据从 MISO 脚移出发送到主机的 MISO 脚。

传输完一个字节后，SPI 时钟发生器停止，传输完成标志（SPIF）置位并产生一个中断（如果 SPI 中断使能（ESPI, 或 IEN1.3＝1））。主机和从机 CPU 的两个移位寄存器可以看作是一个 16 循环移位寄存器。当数据从主机移位传送到从机的同时，数据也以相反的方向移入。这意味着在一个移位周期中，主机和从机的数据相互交换。

12.4 通过$\overline{SS}$改变模式

如果 SPEN=1, SSIG=0 且 MSTR=1，SPI 使能为主机模式。$\overline{SS}$脚可配置为输入（P2M2.4, P2M1.4＝00）或准双向模式（P2M2.4, P2M1.4＝01）。这种情况下，另外一个主机可将该脚驱动为低电平，从而将该器件选择为 SPI 从机并向其发送数据。为了避免争夺总线，SPI 变成从机。SPI 变成从机后，MOSI 和 SPICLK 管脚强制变为输入模式，MISO 则变为输出模式。

SPSTAT 的 SPIF 标志置位。如果 SPI 中断使能，则产生 SPI 中断。

用户软件必须一直对 MSTR 位进行检测，如果该位被一个从机选择所清零而用户想继续将 SPI 作为主机，这时就必须重新置位 MSTR，否则就进入从机模式。

12.5 写冲突

SPI 在发送时为单缓冲，在接收时为双缓冲。这样在前一次发送尚未完成之前，不能将新的数据写入移位寄存器。当发送过程中对数据寄存器进行写操作时，WCOL 位（SPSTAT.6）将置位以指示数据冲突。这种情况下，当前发送的数据继续发送，而新写入的数据（即造成冲突的数据）将丢失。

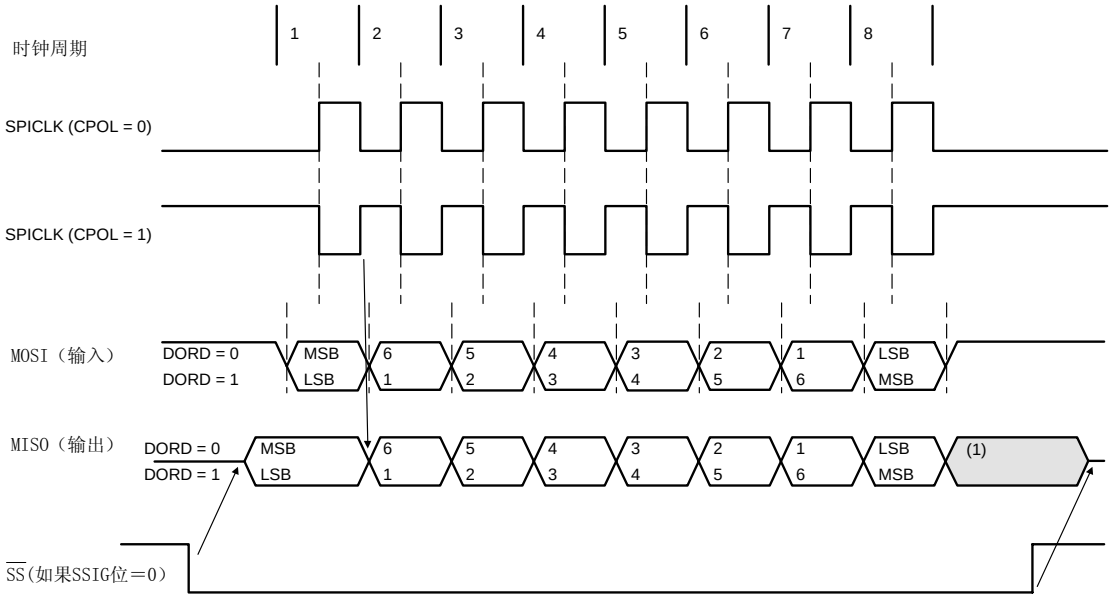
当对主机或从机进行写冲突检测时，主机发生写冲突的情况是很罕见的，因为主机拥有数据传输的完全控制权。但从机有可能发生，因为当主机启动传输时，从机无法进行控制。

接收数据时，接收到的数据传送到一个并行读数据缓冲区，这样将释放移位寄存器以进行下一个数据的接收。但必须在下个字符完全移入之前从数据寄存器中读出接收到的数据，否则，前一个接收数据将丢失。

WCOL 可通过软件向其写入“1”清零。

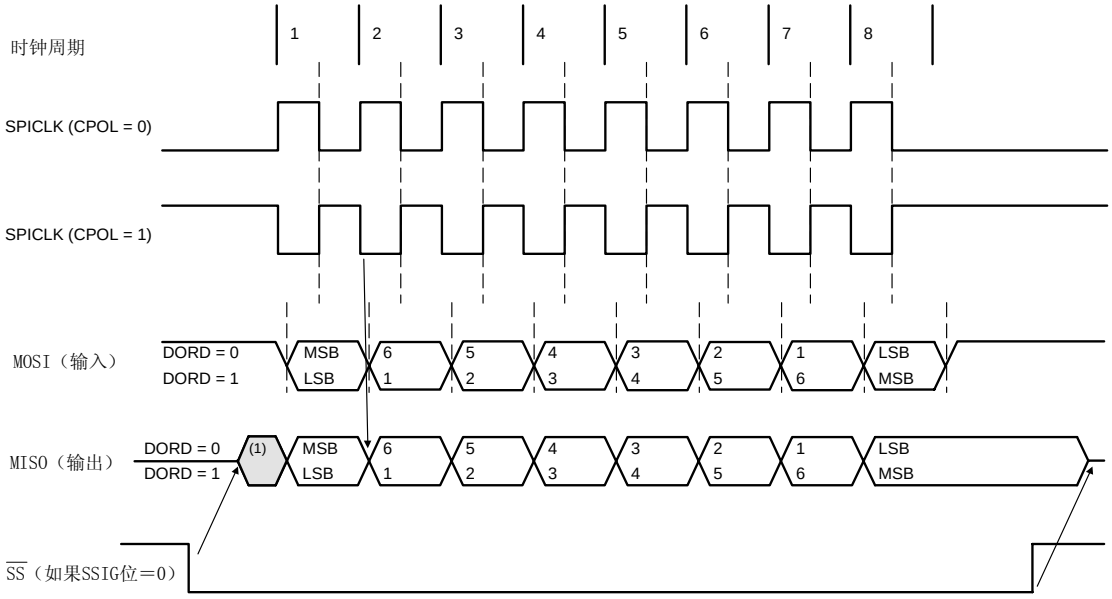
12.6 数据模式

时钟相位位(CPHA)允许用户设置采样和改变数据的时钟边沿。时钟极性位 CPOL 允许用户设置时钟极性。图 36～图 39 所示为时钟相位位 CPHA 的不同设定。



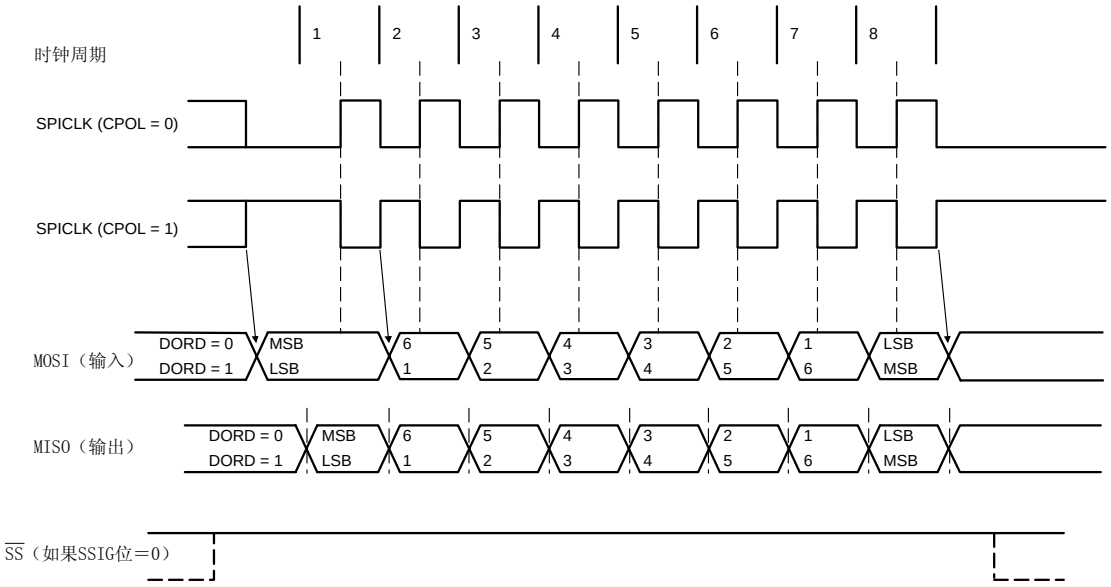
（1） 未定义

图 36　SPI 从机传输格式（CPHA=0）



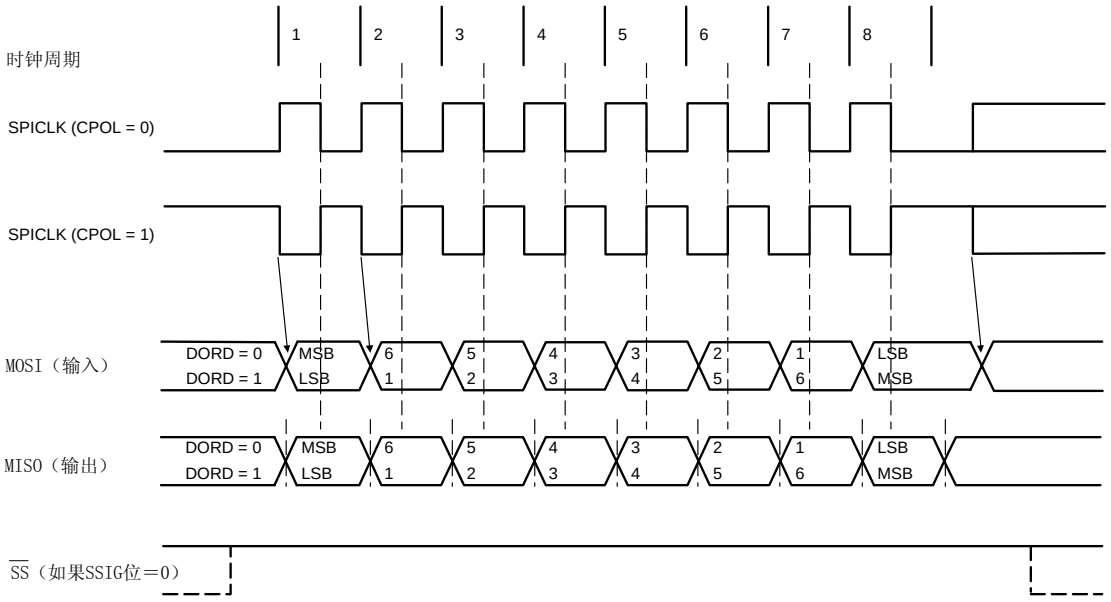
（1）未定义

图 37　SPI 从机传输格式（CPHA=1）



（1）未定义

图 38　SPI 主机传输格式（CPHA=0）



（1）未定义

图 39　SPI 主机传输格式（CPHA=1）

12.7 SPI 时钟预分频器选择

SPI 时钟预分频器选择是通过 SPCTL 寄存器中的 SPR1-SPR0 位实现的（见表 76）。

13.模拟比较器

P89LPC952 有 2 个模拟比较器，输入和输出选项允许将比较器配置成为不同模式。当正向输入(二个可选择脚之一)电压大于反向输入时(可选择外部管脚输入或内部参考电压)，输出信号为“1”(可从寄存器读出和/或输出到管脚)，反之则输出为“0”。每个比较器都可配置为当输出值发生改变时产生中断。

13.1 比较器配置

每个比较器都有各自的控制寄存器。比较器 1 为 CMP1，比较器 2 为 CMP2。两个控制寄存器完全相同，如表 82 所示。

两个比较器全部的连接方式如图 40 所示。每个比较器有 8 种可能的配置方式，由对应 CMPn 寄存器的控制位：CPn,CNn 和 OEn 来决定，如图 41 所示。

当每个比较器刚被使能时，比较器输出和中断标志需要 10 微秒的稳定时间，在这段时间里，相应的比较器中断不应使能，并且在使能中断以前必须清零相应的比较器中断标志，以避免立即响应中断服务。

表 81　比较器控制寄存器（CMP1-地址 ACh，CMP2-地址 ADh）位分配

位	7	6	5	4	3	2	1	0
符号	-	-	CEn	CPn	CNn	OEn	COn	CMFn
复位	x	x	0	0	0	0	0	0

表 82　比较器控制寄存器（CMP1-地址 ACh，CMP2-地址 ADh）位描述

位	符号	描述
0	CMFn	比较器中断标志。当比较器输出 COn 状态改变时由硬件置位。使能比较器中断时，该位置位可产生硬件中断。通过软件清零。
1	COn	比较器输出，与 CPU 时钟同步以允许软件进行读取。
2	OEn	输出使能，如果比较器使能（CEn＝1），则该位为 1 时比较输出结果连接到 CMPn 管脚。此输出和 CPU 时钟不同步。
3	CNn	比较器反向输入选择，为‘0’时选择 CMPREF 作为比较器反向输入，为‘1’时选择内部比较器参考电压 Vref 作为比较器反向输入。
4	CPn	比较器正向输入选择，为‘0’时选择 CINnA 作为正向输入，为‘1’时选择 CINnB 作为正向输入。
5	CEn	比较器使能位。当置位时，对应的比较器使能。CEn 置位 10 微秒后比较器输出有效值。
6:7	-	保留

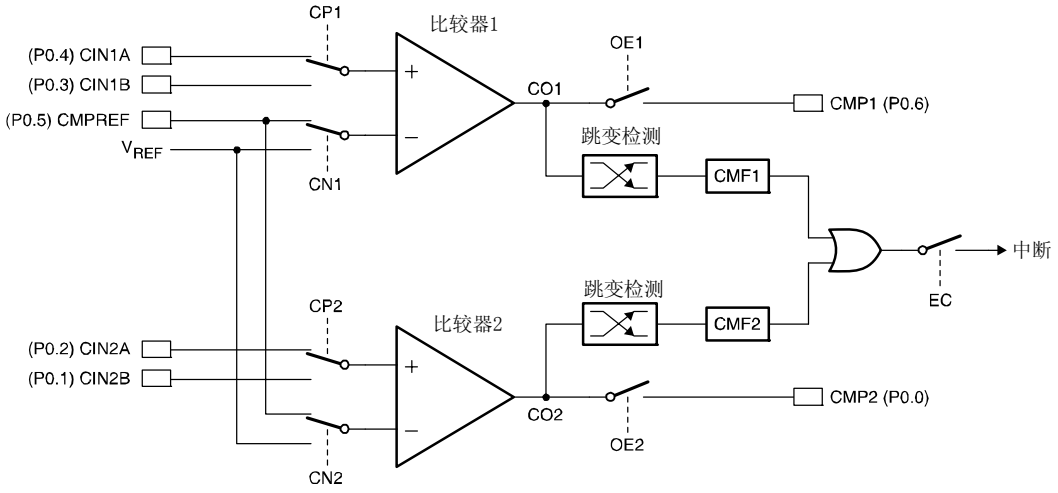


图 40　比较器输入和输出的连接

13.2 内部参考电压

当使用比较器的单个输入管脚时，内部参考电压 Vref 可提供一个默认的参考电压。请情请参考 P89LPC952 数据手册。

13.3 比较器输入管脚

比较器输入和参考管脚可用作数字 I/O 口或比较器模拟输入。当用作数字 I/O 口时这些管脚可承受 5V 的电压。但是，当用作比较器 CMPn 的输入信号时，必须满足低电压的限制。请参考 P89LPC952 数据手册。

13.4 比较器中断

每个比较器配置寄存器中都有一个比较器中断标志位 CMFn。当比较器输出状态改变时中断标志位置位，此标志位可通过软件查询或用于产生一个中断。两个比较器使用同一个中断向量。如果 IEN1 寄存器的中断使能位 EC 置位且通过配置 IEN0 寄存器的 EA 位使中断系统使能，可产生中断。如果两个比较器都使能中断，在进入中断服务程序之后，用户需要读取中断标志以确定是哪一个比较器产生的中断。

当比较器被禁止时，比较器的输出 COx 变为高电平。如果比较器在输出为低电平时被禁止，比较器的输出就由低变高，这样将使比较器标志 CMFx 置位。如果比较器中断被使能，将产生中断。因此，用户必须在比较器被禁止前关闭比较器中断，另外，用户应在比较器被禁止后清除比较器中断标志 CMFx。

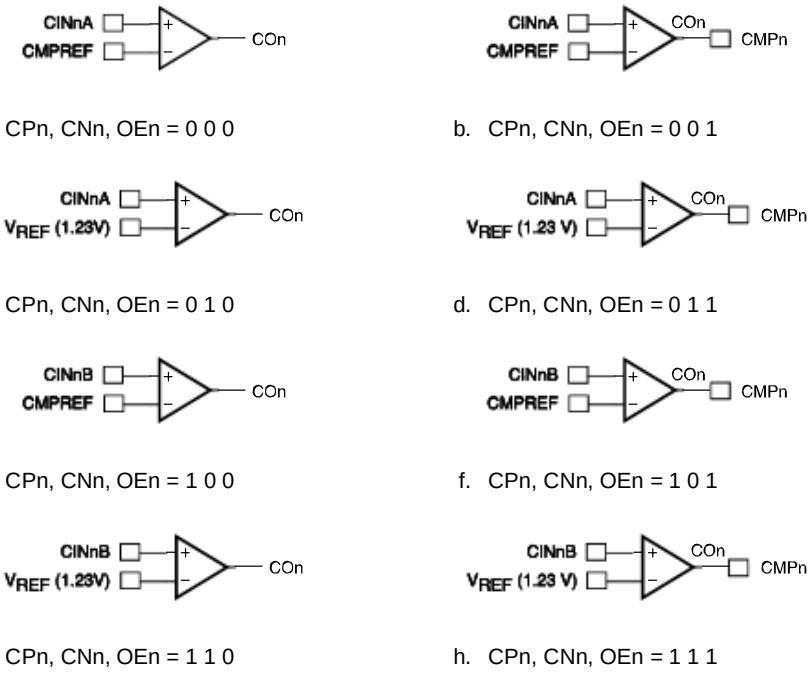
13.5 比较器和节电模式

在掉电模式或空闲模式下，两个比较器或其中一个可以继续保持使能状态。但在完全掉电模式中，两个比较器都被自动禁止。

当比较器中断使能时（完全掉电模式除外），比较器输出发生改变时将会产生一个中断并将处理器唤醒。当比较器输出到管脚使能时，此管脚应该配置为推挽输出模式以便在掉电

工作模式下获得较短的转换时间。这样做是因为当振荡器停止后，接通准双向口不会产生正常情况下的短时强上拉。

比较器在掉电或空闲状态下的功耗和正常操作模式下相同。当系统功耗是一个重要的指标时，就必须将比较器的功耗考虑在内。若要降低功耗，用户可通过置位 PCONA.5 来禁止比较器，或将器件设置为完全掉电模式。



a. CPn, CNn, OEn = 0 0 0　　b. CPn, CNn, OEn = 0 0 1

c. CPn, CNn, OEn = 0 1 0　　d. CPn, CNn, OEn = 0 1 1

e. CPn, CNn, OEn = 1 0 0　　f. CPn, CNn, OEn = 1 0 1

g. CPn, CNn, OEn = 1 1 0　　h. CPn, CNn, OEn = 1 1 1

图 47　比较器配置

13.6 比较器配置举例

下面是一段比较器初始化的程序。比较器 1 配置如下：CIN1A 和 CMPREF 输入，输出到 CMP1 管脚，比较器输出改变时产生中断。

```
CMPINIT:
    MOV     PT0AD,#030h     ；禁止管脚 CIN1A,CMPREF 的数字输入功能
    ANL     P0M2,#0CFh      ；禁止 CIN1A,CMPREF 的数字输出功能
    ORL     P0M1,#030h      ；CIN1A, CMPREF 使用模拟功能
    MOV     CMP1,#024h      ；开启比较器 1，并进行如下设置：
                            ；－CIN1A 为正向输入。
                            ；－CMPREF 管脚为反向输入。
                            ；－比较结果输出到 CMP1。
    CALL    delay10us       ；比较器启动后至少 10us 方可使用
    ANL     CMP1,#0FEh      ；清除比较器 1 的中断标志。
    SETB    EC              ；使能比较器中断。
    SETB    EA              ；使能中断系统（如果有必要）
    RET                     ；中断返回
```

比较器中断服务程序返回前必须先清除中断标志（此例中为 CMF1）。

14.键盘中断（KBI）

键盘中断功能主要用于当 P0 口等于或不等于特定的模式时产生一个中断。该功能可用于总线地址识别或对键盘的识别。用户可通过 SFR 将端口配置为不同的用途。

有 3 个特殊功能寄存器用于该功能。键盘中断屏蔽寄存器（KBMASK）用于定义连接到 P0 口的使能触发中断的输入管脚。键盘模式寄存器（KBPATN）用于定义与 P0 口值相比较的模式。当键盘中断功能有效且条件匹配时，键盘中断控制寄存器（KBCON）中的键盘中断标志（KBIF）置位。如果通过置位 IEN1 寄存器的 EKBI 位和 EA 位将中断使能，则会产生一个中断。键盘中断控制寄存器（KBCON）中的 PATN_SEL 位用于定义比较的等于或不等于。

为了将键盘中断设置为类似 87LPC76x 的 KBI 功能，用户必须设置 KBPATN=0FFH 和 PATN_SEL=0（不相等），这样由 KBMASK 寄存器使能的任何连接到 P0 口管脚按键都将使硬件置位 KBIF 并产生中断（如果中断使能）。中断可用于将 CPU 从空闲模式或掉电模式中唤醒。此特性尤其适合便携式且使用电池供电的系统，因为这些系统需要对功耗进行管理同时又要方便用户使用。

为了置位中断标志并导致中断产生，P0 口的模式的保持时间必须长于 6 个 CCLK。

表 83　键盘模式寄存器（KBPATN-地址 93h）位分配

位	7	6	5	4	3	2	1	0
符号	KBPATN.7	KBPATN.6	KBPATN.5	KBPATN.4	KBPATN.3	-KBPATN.2	KBPATN.1	KBPATN.0
复位	1	1	1	1	1	1	1	1

表 84 键盘模式寄存器（KBPATN-地址 93h）位描述

位	符号	访问	描述
0～7	KBPATN.7	R/W	模式位 0～7

表 85　键盘控制寄存器（KBCON-地址 94h）位分配

位	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	-	PATN_SEL	KBIF
复位	x	x	x	x	x	x	0	0

表 86　键盘控制寄存器（KBCON-地址 94h）位描述

位	符号	访问	描述
0	KBIF	R/W	键盘中断标志。当 P0 口的值与用户在 KBPATN, KBMASK 和 PATN_SEL 中定义的条件相匹配时，该位置位。需要通过软件向其写入 0 清零。
1	PATN_SEL	R/W	模式匹配极性选择。该位置位时，P0 口必须等于用户在 KBPATN 中定义的模式才能产生中断。清零时，P0 口必须不等于 KBPATN 寄存器的值时才能产生中断。
2:7	-	-	保留。

表 87　键盘中断屏蔽寄存器（KBMASK-地址 86h）位分配

位	7	6	5	4	3	2	1	0
符号	KBMASK.7	KBMASK.6	KBMASK.5	KBMASK.4	KBMASK.3	KBMASK.2	KBMASK.1	KBMASK.0
复位	0	0	0	0	0	0	0	0

表 88　键盘中断屏蔽寄存器（KBMASK-地址 86h）位描述

位	符号	描述
0	KBMASK.0	该位置位时，使能 P0.0 作为键盘中断源
1	KBMASK.1	该位置位时，使能 P0.1 作为键盘中断源
2	KBMASK.2	该位置位时，使能 P0.2 作为键盘中断源
3	KBMASK.3	该位置位时，使能 P0.3 作为键盘中断源
4	KBMASK.4	该位置位时，使能 P0.4 作为键盘中断源
5	KBMASK.5	该位置位时，使能 P0.5 作为键盘中断源
6	KBMASK.6	该位置位时，使能 P0.6 作为键盘中断源
7	KBMASK.7	该位置位时，使能 P0.7 作为键盘中断源

[1] 为了使 KBMASK 寄存器的设置有效，键盘中断必须被使能。

15.看门狗定时器(WDT)

看门狗定时器子系统可通过复位使系统从错误的代码操作中恢复。当软件没能在定时器到达终端计数之前将其重装而产生计数下溢时，看门狗定时器就会引起系统复位。看门狗定时器只能通过上电复位实现复位。

15.1 看门狗功能

WDCON 和 UCFG1 寄存器控制着 WDT 的开/关、WDT 的时钟源、预分频值以及溢出时 WDT 是否能使器件复位。另外，WDT 还含有一种安全机制，通过 IAP 或商业编程器写入 UCFG1 的配置值能强制 WDT 进入使能状态。

如果 WDTE 位（UCFG1.7）被置位，使能 WDT 溢出时对器件的复位。复位后，不管 WDTE 位是何种状态，WDT 将继续运行。

通过 WDRUN 位（WDCON.2）的置位和清零来分别起动和关闭 WDT。复位后该位被置位，WDT 保持运行状态。每次写 WDCON 操作后需跟随一个清零序列（见 15.2 节）。用户可以通过 WDCON 的其它位来选择 WDT 的时钟源和预分频值。

当未使能看门狗溢出对器件的复位时，WDT 可工作在“定时器模式”中，如果需要可使能产生中断（IEN0.6）。

看门狗安全使能位 WDSE（UCFG1.4）和 WDTE 一起，可在上电时使看门狗强制进入特定的工作状态。详情请参考表 89。

图 44 所示为看门狗模式中的看门狗定时器。它包含一个可编程的 13 位预分频器和一个 8 位倒计数器。该倒计数器以预分频器的节拍作为时钟源（递减）。预分频器的时钟源可选

择 PCLK 或看门狗振荡器，由 WDCON 寄存器的 WDCLK 位决定。（注：时钟源的切换不会立即生效—见 15.3 节。）

当看门狗复位使能且看门狗产生计数下溢出时，看门狗将请求执行复位。当看门狗复位使能时，在写入 WDL 或 WDCON 后必须执行一个看门狗清零序列以使新的值生效。

如果发生看门狗复位，内部复位至少保持一个看门狗时钟周期（PCLK 或看门狗振荡器时钟）有效。如果 CCLK 仍然运行，代码将在复位周期之后立即开始执行。如果处理器处于掉电模式，看门狗复位将启动振荡器并在振荡器稳定之后恢复代码的执行。

表 89　看门狗定时器配置

WDTE	WDSE	功能
0	X	看门狗复位禁止。看门狗定时器可作为内部定时器并可产生中断。此时，WDSE 的值无影响。
1	0	看门狗复位使能。用户通过设置 WDCLK 来选择时钟源。
1	1	看门狗复位使能，还具有以下安全特性： 1. WDCLK 强制为 1（使用看门狗振荡器） 2. WDCON 和 WDL 寄存器只可写一次 3. WDRUN 强制为 1

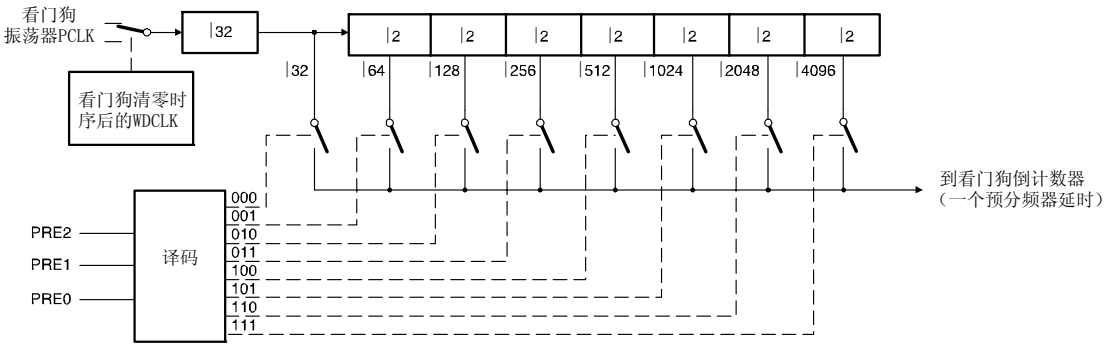


图 42　看门狗预分频器

15.2 看门狗清零序列

看门狗定时器控制寄存器和 8 位倒计数器(图 43)并不直接由用户装载。用户对 WDCON 和 WDL 进行写操作。在清零序列结束时，WDCON 和 WDL 寄存器的值装入控制寄存器和 8 位倒计数器。在清零序列之前，任何写入这两个寄存器的新值都将无效。为了避免看门狗复位，看门狗定时器必须在定时溢出前进行清零（通过一个叫做“清零序列”的特殊软件时序）。

要将看门狗清零，必须按顺序执行两条指令。在这两条写指令之间允许读出 SFR 但不允许写操作。这两条指令就是将 A5H 送入 WFEED1 寄存器和将 5AH 送入 WFEED2 寄存器。不正确的清零序列会立即导致看门狗复位。下面是一个清零序列操作实例：

```
CLR     EA              ; 禁止中断
MOV     WFEED1,#0A5h    ; 执行清零第一部分
MOV     WFEED2,#05Ah    ; 执行清零第二部分
SETB    EA              ; 使能中断
```

该序列是假设 P89LPC952 中断系统已经使能且有可能在看门狗清零的过程中产生中断请求。如果在执行看门狗清零序列中允许响应中断并且该中断服务程序中包含对任意的 SFR 写操作，将会触发看门狗复位。如果确知在清零序列中不会发生中断，可以取消禁止和重新使能中断的指令。

在看门狗模式中（WDTE=1），写入 WDCON 后必须立即执行一次清零序列以使 WDL 装入 8 位倒计数器，WDCON 装入映像寄存器。如果不这样操作将导致看门狗复位。

例如：设置 WDRUN=1：

```
MOV     ACC,WDCON           ; 取出 WDCON 的值
SETB    ACC.2               ; 设置 WD_RUN=1
MOV     WDL,#0FFh           ; 新值装入 8 位倒计数器
CLR     EA                  ; 禁止中断
MOV     WDCON,ACC           ; 写回到 WDCON（在看门狗使能后，必须立即执行清零序列）
MOV     WFEED1,#0A5h        ; 执行清零第一部分
MOV     WFEED2,#05Ah        ; 执行清零第二部分
SETB    EA                  ; 使能中断
```

在定时器模式中（WDTE=0），WDCON 每 CCLK 周期被装入控制寄存器一次（装入控制寄存器时不要求清零序列），但在溢出发生之前将 WDL 寄存器值装入 8 位倒计数器时需要执行清零序列。

看门狗定时器在溢出前的时钟个数可由下面的等式得到：

$$tclks = (2^{(5+PRE)})(WDL+1)+1 \quad (1)$$

其中，

- PRE 为预分频器的值（PRE2-PRE0），范围为 0 到 7。
- WDL 为看门狗装载寄存器的值，范围为 0 到 255。

因此 tclks 的最小个数为：

$$tclks = (2^{(5+0)})(0+1)+1 = 33 \quad (2)$$

tclks 的最大个数为：

$$tclks = (2^{(5+7)})(255+1)+1 = 1{,}048{,}577 \quad (3)$$

表 104 所示为 P89LPC952 看门狗溢出值的取样。

表 90　看门狗定时器控制寄存器（WDCON-地址 A7h）位分配

位	7	6	5	4	3	2	1	0
符号	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK
复位	1	1	1	x	x	1	1/0	1

表 91　看门狗定时器控制寄存器（WDCON-地址 A7h）位描述

位	符号	描述
0	WDCLK	看门狗输入时钟选择。当该位为 1 时，选择看门狗振荡器作为时钟源；为 0 时，选择 PCLK。（如果 CPU 处于掉电状态，当 WDCLK=0 时，看门狗被禁止，见 16.5 节）。（注：如果 WDTE 和 WDSE 都置位，该位被强制为 1。）详细内容请参考 16.3 节。
1	WDTOF	看门狗定时器溢出标志。该位在 8 位倒计数器溢出时置位。在看门狗模式中，一个清零序列将清零该位。还可通过软件向其写入 0 将其清零。
2	WDRUN	看门狗运行控制。看门狗定时器在 WDRUN=1 时启动并在 WDRUN=0 时停止。该位在 WDTE 和 WDSE 都置位时被强制设置为 1（看门狗运行）。
3:4	-	保留
5	PRE0	时钟预分频器节拍选择，详见表 104。
6	PRE1	
7	PRE2	

表 92　看门狗溢出值取样

PRE2-PRE0	WDL (十进制)	溢出时间 (以看门狗时钟周期显示)	看门狗时钟源	
			400KHz 看门狗振荡器时钟（标称值）	12MHz CCLK(6MHz CCLK/2 看门狗时钟)
000	0	33	82.5µs	5.50µs
	255	8,193	20.5ms	1.37 ms
001	0	65	162.5µs	10.8µs
	255	16,385	41.0ms	2.73 ms
010	0	129	322.5µs	21.5µs
	255	32,769	81.9ms	5.46 ms
011	0	257	642.5µs	42.8µs
	255	65,537	163.8ms	10.9 ms
100	0	513	1.28ms	85.5µs
	255	131,073	327.7ms	21.8 ms
101	0	1,025	2.56ms	170.8µs
	255	262,145	655.4ms	43.7 ms
110	0	2,049	5.12ms	341.5µs
	255	524,289	1.31s	87.4 ms
111	0	4097	10.2ms	682.8µs
	255	1,048,577	2.62s	174.8 ms

15.3 看门狗时钟源

看门狗定时器系统包含一个片内 400kHz 振荡器。通过配置 WDCON 的 WDCLK 位来选择看门狗振荡器或 PCLK（参考图 42）作为看门狗定时器的时钟源。如果看门狗特性被使能，为了防止看门狗溢出时复位 CPU，必须定时通过软件将定时器清零。

在改变 WDCLK(WDCON.0)之后，对时钟源的切换不会立即生效。如图 44 所示，要在一个清零序列之后才将选择装入。此外，由于时钟同步逻辑的关系，在放弃旧的时钟源之前还需要经过两个旧的时钟周期。然后需要两个新的时钟周期以使新的时钟有效。

由于清零序列结束后预分频器立刻开始计数，切换时钟会造成预分频器的计数误差。此误差为 2 个旧时钟周期加 2 个新时钟周期。

注：当切换时钟时，很重要的一点就是，在完成清零序列后将旧的时钟再保持 2 个时钟周期。否则，当旧的时钟源禁止时看门狗也被禁止。例如，假设 PCLK(WCLK=0)为当前的时钟源，在将 WCLK 设置为 1 后，程序在清零序列完成后至少应当等待 2 个 PCLK(4 个 CCLK)才能进入掉电模式。否则，当 CCLK 关闭时，看门狗也会被禁止。看门狗振荡器永远也不可能成为时钟源，除非 CCLK 再次打开。

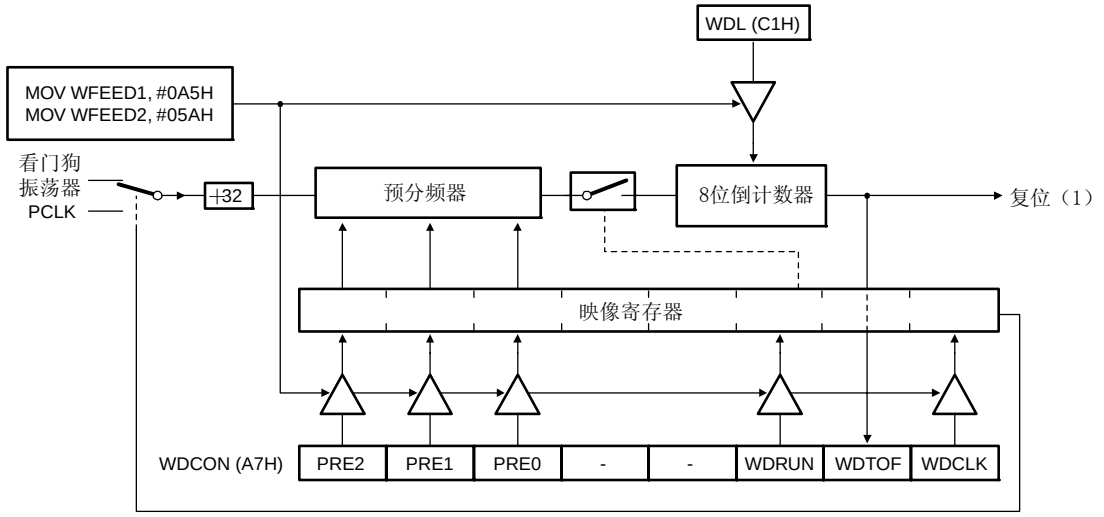


图 43　看门狗模式中的看门狗定时器（WDTE=1）

15.4 定时器模式中的看门狗定时器

图 44 所示为定时器模式中的看门狗定时器。在该模式中，WDCON 的任何改变都将在一个看门狗时钟周期后写入映像寄存器。看门狗溢出将置位 WDTOF 位。如果 IEN0.6 置位，看门狗溢出被使能以产生中断。WDTOF 通过软件向其写入 0 清零。看门狗溢出时，WDL 的内容重新装入倒计数器并且倒计数器立即重新开始计数。

溢出前，将 WDL 装入倒计数器仍然需要执行清零序列。在该模式中错误的清零序列将被忽略。

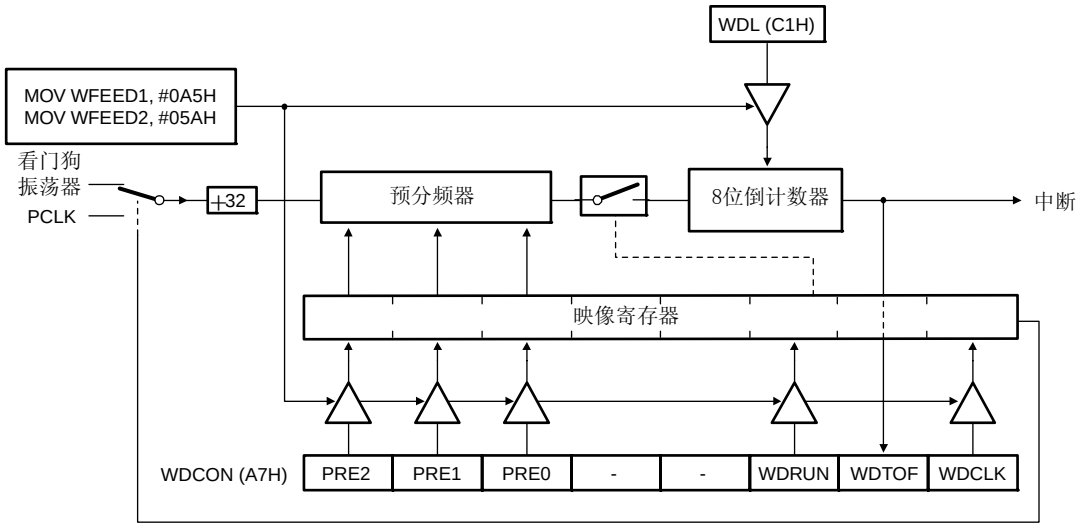


图 44　定时器模式中的看门狗定时器（WDTE=0）

15.5 掉电操作

掉电模式下，只要 WDT 振荡器用作 WDT 的时钟源，WDT 振荡器将继续运行，消耗大约 50uA 的电流。如果选择 PCLK 作为 WDT 的时钟源，WDT 振荡器将和器件的其它部分一同进入掉电模式（见 15.3 节）。掉电模式下 PCLK 停止运行，导致看门狗被禁止。

15.6 无外部振荡器的掉电周期唤醒

在不使用外部振荡器的条件下，周期唤醒所需功耗取决于用来产生唤醒信号的内部振荡源的功耗。采用内部 RC 振荡器的实时时钟可以用作唤醒源，该振荡器消耗的电流大约为 300uA。如果用 WDT 代替 RC 振荡器来产生中断，消耗的电流可降至 50uA。WDT 溢出时可将器件唤醒。

16.附加特性

AUXR1 寄存器包含几个特殊用途的控制位，这些位分别与几种芯片功能相关联。AURX1 如表 94 所示。

表 93　AUXR1 寄存器（地址 A2h）位分配

位	7	6	5	4	3	2	1	0
符号	CLKLP	EBRR	ENT1	ENT0	SRST	0	-	DPS
复位	0	0	0	0	0	0	x	0

表 94　AUXR1 寄存器（地址 A2h）位描述

位	符号	描述
0	DPS	数据指针选择。选择两个数据指针中的一个。
1	-	未用。允许设置成“1”。
2	0	该位包含一个硬件 0。这样就允许通过将 AUXR1 加 1 来使 DPS 位翻转而不影响寄存器的其它位。
3	SRST	软件复位。当 SRST 通过软件置位时，P89LPC952 就像硬件复位一样复位。
4	ENT0	置位时，定时器 0 溢出使 P1.2 翻转。输出频率是定时器 0 溢出速率的一半。详情请参考第八节“定时器 0 和 1”。
5	ENT1	置位时，定时器 1 溢出使 P0.7 翻转。输出频率是定时器 1 溢出速率的一半。详情请参考第八节“定时器 0 和 1”。
6	EBRR	UART 间隔检测复位使能。如果该位为 1，UART 间隔检测将导致芯片复位并强制器件进入 ISP 模式。
7	CLKLP	时钟低功耗选择。置位时降低时钟电路的功耗。当时钟频率为 8MHz 或以下时可置位该位。复位后该位清零支持时钟频率高达 12MHz 的操作。

16.1 软件复位

AUXR1 寄存器的 SRST 位使软件能像发生外部复位或看门狗复位一样，使处理器彻底复位。当位 3 为 1 的数据被写入 AUXR1 时，所有 SFR 将被初始化，程序重新从地址 0000 处开始执行。因此，写入 AUXR1 时务必小心以防止产生意外的软件复位。

16.2 双数据指针

双数据指针（DPTR）结合特定的指令增加了处理器指向地址的方式。AUXR1 寄存器的 DPS 位选择两个数据指针中的一个。软件无法访问当前未被选择的 DPTR，除非 DPS 位状态改变。

受数据指针选择影响的指令有：

- INC　　DPTR　　数据指针加 1
- JMP　　@A+DPTR　　跳转到 DPTR 的相对间接地址
- MOV　　DPTR,#data16　　数据指针装入一个 16 位的常数
- MOVC　　A,@A+DPTR　　将 DPTR 相对地址的代码字节移入累加器
- MOVX　　A,@DPTR　　将 DPTR 指向的数据存储单元的数据字节移入累加器
- MOVX　　@DPTR,A　　将累加器的数据字节移入 DPTR 指向的数据存储单元

而且任何读取或操纵 DPH 和 DPL 寄存器（当前 DPTR 的高字节和低字节）的指令都受 DPS 位设置的影响。MOVX 指令在 P89LPC952 中的应用受到限制，因为该器件没有外部数据总线。但是，它们可用来访问 Flash 配置信息（见“Flash 配置”）或辅助数据（XDATA）存储器。

AUXR1 的位 2 永远都是逻辑 0。这样 DPS 位可以简单地通过将 AUXR1 加 1 来翻转（由此来切换数据指针），而不会改变寄存器的其它位。

17.Flash 程序存储器

17.1 概述

P89LPC952 Flash 存储器提供在电路电擦除和编程。Flash 可以字节为单位进行读取和写入。扇区和页擦除功能可擦除任何 Flash 扇区（1kB）或页（64 字节）。芯片擦除操作可擦除整个程序存储器。P89LPC952 提供了 5 种编程方法。片内产生的擦除和写入时序为用户提供了友好的编程接口。P89LPC952 Flash 存储器甚至在经过 100, 000 次擦除和编程周期之后仍然能可靠地保存存储器的内容。存储单元的设计优化了擦除和编程结构。P89LPC952 使用 V_{DD} 电压来执行编程/擦除算法。

17.2 特性

- 使用工业标准的商用编程器进行并行编程。
- 使用工业标准的商用编程器进行在电路串行编程（ICP）。
- 在终端应用程序控制下，IAP-Lite 允许代码存储器的单个或多个字节用作数据存储和编程。
- 内部固化的引导 ROM，包含了一个低级的在应用编程(IAP)子程序，可被终端应用程序调用（除了 IAP-Lite 以外）。
- 默认的串行装载程序可通过串口进行在系统（ISP）编程。该程序位于用户程序存储器空间的顶端。
- Boot 向量允许将用户提供的 Flash 装载代码放入 Flash 存储器内的任何位置。这种配置为用户提供了极大的灵活性。
- 可在整个工作电压范围内执行编程和擦除。
- 使用 ISP/IAP/IAP-Lite 进行读/编程/擦除。
- 任意 Flash 编程/擦除时间在 2ms 内（4ms 擦除/编程）。
- 可对 Flash 的代码以扇区为单位进行编程加密。
- 每个字节至少可执行 100,000 次擦除/编程。
- 数据至少可保存 10 年。

17.3 Flash 编程和擦除

P89LPC952 包含 1KB 的扇区。每个扇区可进一步分成 64 字节的页。除了扇区擦除和页擦除外，还包含一个 64 字节页寄存器以允许对给定页的 1 到 64 字节同时进行编程，这样可彻底降低整个编程的时间。P89LPC952 提供了 5 种编程方法：

- 利用工业标准的商用编程器进行并行编程。
- 利用工业标准的商业编程器进行在电路串行编程（ICP）。
- 在终端应用程序控制下，IAP-Lite 允许代码存储器的单个或多个字节用于数据存储和编程。

- 内部固定的引导 ROM，包含了一个低级的在应用编程(IAP)子程序，可被终端应用程序调用（除 IAP-Lite 以外）。
- 出厂提供的默认串行装载程序，位于用户程序存储器空间的顶端，通过串口提供在系统编程（ISP）。

17.4 Flash 用作数据存储器：IAP-Lite

除标准 IAP 功能外，P89LPC952 的 Flash 程序存储阵列还支持 IAP-Lite 功能。程序存储阵列中未加密扇区的任何字节都使用 MOVC 指令执行读操作，因此，这些字节可用作非易失性数据存储。IAP-Lite 提供的擦除编程功能，可方便地使用一条指令完成一页中的一个或多个字节的擦除和编程，而不影响该页中其它字节的内容。在微控制器固件的控制下，IAP-Lite 在应用程序中利用 4 个 SFR 和内部 64 字节“页寄存器”对未加密扇区内的字节执行擦除和编程操作。这 4 个 SFR 为：

- FMCON（Flash 控制寄存器）。读时用作状态寄存器；写时用作命令寄存器。注：写入命令时状态位被清零。
- FMADRL，FMADRH（Flash 存储器地址低字节，Flash 存储器地址高字节）。用来指示页寄存器的字节地址或用户程序存储器的页。
- FMDATA（Flash 数据寄存器）。存放要装入页寄存器的数据。

页寄存器包括 64 个字节，每个字节都有一个更新标志。当将一个 LOAD 命令写入 FMCON 时，页寄存器和所有更新标志都被清零。当向 FMDATA 写入数据后，写入的值存放在 FMADRL 低 6 位指向的页寄存器单元。同时，相应单元的更新标志置位。FMADRL 自动增加到下一个单元。在页寄存器的最后一个字节被写入后，FMADRL 重新指向页寄存器的第一个字节，但不影响 FMADRL[7:6]的值。此时，装入页寄存器中的字节不一定是连续的。通过在写入 FMDATA 前改变 FMADRL 的内容可将任何字节单元装入页寄存器。但是，每发出一个 LOAD 命令，页寄存器的每个单元只能被写入一次。不要试图对一个页寄存器单元执行多次写操作。

FMADRH 和 FMADRL[7:6]用来选择执行擦除-编程功能的程序存储器页。当向 FMCON 写入擦除-编程命令后，页寄存器中被更新单元对应在程序存储器页的单元的所有内容被清除，页寄存器中相应单元的内容被编程到程序存储器页中。用户程序阵列中只有装入页寄存器的字节才可被擦除和编程，其它字节不受影响。

向 FMCON 写入擦除-编程命令（68H）将起动擦除-编程处理并使 CPU 进入编程-空闲状态。CPU 会一直保持这种空闲状态直到擦除-编程周期结束或由一个中断终止。从编程-空闲状态退出后，FMCON 的内容是该周期的状态信息。

如果在擦除/编程周期内有中断产生，擦除/编程操作被终止，FMCON 的 OI 标志（操作被中断）置位。如果应用程序允许在擦除-编程过程中产生中断，则每次擦除-编程操作后应用程序将检查 OI 标志（FMCON.0）来判断操作是否被终止。一旦擦除-编程操作被终止，用户程序需要从装载页寄存器开始重复整个过程。

不管有多少个字节装入页寄存器，擦除-编程周期的时间都为 4ms（2ms 擦除，2ms 编程）。

程序存储器中一个字节（或多个字节）的擦除-编程操作步骤如下：

- 向 FMCON 写入 LOAD 命令（00H）。LOAD 命令将清除页寄存器的所有单元及相应的更新标志。

- 将页寄存器内的地址写入 FMADRL。由于 FMADRL[5:0]用来装载页寄存器地址，FMADRH 和 FMADRL[7:6]用于擦除-编程命令，所以，此时用户可对页寄存器内的字节单元（FMADRL[5:0]）和程序存储器页地址（FMADRH 和 FMADRL[7:6]）执行写操作。
- 将要编程的数据写入 FMDATA。这可让 FMADRL 加 1，指向页寄存器的下一个字节。
- 如果需要的话，将要编程的下个字节的地址写入 FMADRL（如果是连续字节就不需要，因为 FMADRL 是自动递增的）。所有将编程的字节必须在同一页中。
- 将要编程的下一字节的数据写入 FMDATA。
- 重复 FMADRL 和/或 FMDATA 的写操作，直至所有要编程的字节都被装入页寄存器。
- 将用户程序存储器的页地址写入 FMADRH 和 FMADRL[7:6]。（如果之前向 FMADRL[5:0]写入页寄存器地址时，没有将用户程序存储器的页地址写入 FMADRH 和 FMADRL[7:6]）。
- 向 FMCON 写入擦除-编程命令（68H），起动擦除-编程周期。
- 读 FMCON 来检查状态。如果操作被终止，启动 LOAD 命令来重复操作。

表 95　Flash 存储器控制寄存器（FMCON-地址 E4h）位分配

位	7	6	5	4	3	2	1	0
符号(R)	-	-	-	-	HVA	HVE	SV	OI
符号(W)	FMCMD.7	FMCMD.6	FMCMD.5	FMCMD.4	FMCMD.3	FMCMD.2	FMCMD.1	FMCMD.0
复位	0	0	0	0	0	0	0	0

表 96　Flash 存储器控制寄存器（FMCON-地址 E4h）位描述

位	符号	访问	描述
0	OI	R	操作被中断。由于中断或复位而使擦除-编程终止时该位置位。
	FMCMD.0	W	命令字节位 0。
1	SV	R	安全出错。试图对加密扇区或页进行编程、擦除或 CRC 操作而产生中断时该位置位。
	FMCMD.1	W	命令字节位 1。
2	HVE	R	高电压错误。高电压发生器出错时该位置位。
	FMCMD.2	W	命令字节位 2。
3	HVA	R	高电压终止。当如果在编程或擦除周期内检测到中断或掉电以及在编程或擦除周期开始时掉电检测被禁止，该位置位。
	FMCMD.3	W	命令字节位 3。
4:7	-	R	保留。
4:7	FMCMD.4	W	命令字节位 4。
4:7	FMCMD.5	W	命令字节位 5。
4:7	FMCMD.6	W	命令字节位 6。
4:7	FMCMD.7	W	命令字节位 7。

装载页寄存器和执行一次擦除/编程操作的汇编程序如下所示。

```
;*************************************************
;*编程用户代码      *
;*************************************************
;*                                               *
;* 输入：                                        *
;*        R3 = 编程字节数（字节）                *
;*        R4 = 页地址 MSB（字节）                *
;*        R5 = 页地址 LSB（字节）                *
;*        R7 = RAM 数据缓冲区指针（字节）        *
;* 输出：                                        *
;*        R7 = 状态（字节）                      *
;*        C = 无错时清零，有错时置位             *
;*************************************************
LOAD    EQU     00H
EP      EQU     68H
PGM_USER:
        MOV     FMCON,#LOAD             ；装载命令，清除页寄存器
        MOV     FMADRH,R4               ；写入高地址
        MOV     FMADRL,R5               ；写入低地址
        MOV     A,R7                    ；
        MOV     R0,A                    ；指针装入 R0
LOAD_PAGE:
        MOV     FMDAT,@R0               ；数据写入页寄存器
        INC     R0                      ；指向下一个字节
        DJNZ    R3,LOAD_PAGE            ；一直操作直至编程字节数为零
        MOV     FMCON,#EP               ；否则擦除并编程页

        MOV     R7,FMCON                ；备份返回的状态
        MOV     A,R7                    ；读状态
        ANL     A,#0FH                  ；保存低四位
        JNZ     BAD                     ；
        CLR     C                       ；如果操作无误，清除错误标志
        RET                             ；返回
BAD:
        SETB    C                       ；设置错误标志
        RET                             ；返回
```

装载页寄存器和执行一次擦除/编程操作的 C 语言程序如下所示。

```
#include <REG952 .H>
unsigned  char  idata  dbytes [64];                    //数据缓冲区
unsigned  char  Fm_stat;                               //状态结果
bit  PGM_USER (unsigned  char , unsigned  char );
bit  prog_fail;

void  main ( )
{
prog_fail = PGM_USER ( 0x1F ,0xC0 );
}

bit  PGM_USER (unsigned  char  page_hi , unsigned  char  page_lo )
     {
          # define  LOAD     0x00                    // 清除页寄存器，使能装载
          # define  EP       0x68                    // 擦除和编程页
          unsigned  char     i;                      // 循环计数
           FMCON = LOAD;                             // 装载命令，清除页寄存器
           FMADRH = page_hi;                         //
           FMADRL = page_lo;                         // 页地址写入地址寄存器
           for ( i= 0 ; i <64 ; i=i+1 )
          {
          FMDATA  = dbytes [i] ;
           }
           FMCON = EP;                          // 擦除和编程页命令
           Fm_stat = FMCON;                     // 读结果状态
           if  ( ( Fm_stat & 0xF ) ! =0 )      prog_fail = 1 ; else  prog_fail = 0;
           return (prog_fail);
           }
```

17.5 在电路编程（ICP）

在电路编程方法允许在没有将微控制器从系统中移出的情况下使用商业编程器来实现编程和擦除功能。在电路编程特性包含了一系列内部硬件资源，通过两线串行接口对 P89LPC952 进行远程编程。PHILIPS 已为嵌入式应用提供了在电路编程特性，最大限度减少了额外的元件开销和电路板面积。ICP 功能使用芯片的 5 个管脚（V_{DD},V_{SS},P0.5,P0.4 和 $\overline{RST}$ ）。使用该特性时只需要一个小的连接器将您的应用电路和外部编程器相连。

17.6 P89LPC952 的 ISP 和 IAP 性能

在应用中编程（IAP）接口允许用户的终端应用程序对用户程序存储器进行擦除和重新编程。此外，还支持对用户可编程字节，包括 UCFG1、引导状态位和引导向量的擦除和重新编程。芯片出厂后，用户代码空间的高 512 字节包含了一个串行在系统编程（ISP）装载程序，允许器件通过串口实现在电路编程。ISP 引导装载程序通过公共的入口依次调用用户

应用程序也可使用的低级子程序。

17.7 引导 ROM

微控制器包含一个 256 字节的引导 ROM，与用户的 Flash 存储器独立。引导 ROM 中的程序用来对擦除和编程用户 Flash 存储器过程中的所有细节操作进行处理。用户程序只需使用适当的参数调用引导 ROM 中的公共入口即可实现所需要的操作。引导 ROM 的操作包括以下内容：擦除扇区，擦除页，编程页，CRC，编程保密位等等。引导 ROM 占用 FF00 到 FFFFH 的程序存储器顶部的地址空间，因此不与用户程序存储器空间冲突。该功能与 IAP-Lite 不同。

17.8 上电复位代码的执行

P89LPC992 包含两个特殊的 Flash 单元：引导向量和引导状态位。复位后，P89LPC952 检查引导状态位的内容。如果为 0，则转去 0000H 地址开始执行程序。这是用户应用代码的正常起始地址。如果引导状态位为 1，则将引导向量的值作为程序计数器的高字节，低字节固定为 00H。

P89LPC952 的出厂默认设定值如下表 111 所示。

用户可擦除工厂预先编好的引导程序。用户如果需要使用这部分代码，就必须小心以避免将器件最后的 1KB 扇区擦除。而采用页擦除功能擦除该扇区的 8 个 64 字节页。如果需要的话，可以将引导向量设置到用户的引导装载程序入口地址。

表 97　引导装载程序的地址和默认引导向量

器件	Flash 空间	截止地址	标识字节			向量大小	页规格	预编程串行装载程序	默认引导向量
			Mfg.id	Id 1	Id2				
P89LPC952	8kB×8	1FFFh	15h	DDh	25h	1kB×8	64×8	1E00h-1FFFh	1Fh

17.9 引导装载程序的硬件激活

在上电时，通过强制器件进入 ISP 模式也可执行引导装载程序（见图 45）。在器件上电时使复位脚保持低电平，并在 V_{DD} 上升到其正常的工作电压之后使复位脚的低电平持续一个固定时间，然后，再给复位管脚提供 3 个准确定时的低电平脉冲。脉冲数目只能是 3 个，少于或多于 3 个脉冲都会导致器件无法进入 ISP 模式。相关时序请参考器件的数据手册。

它等效于引导状态位的值非零。这样就可以在正常执行用户代码时进行手动操作强制器件进入 ISP 模式。如果改变引导向量出厂时的默认设定值，将不再指向工厂预先编程的 ISP 引导装载程序。此时，改变引导向量的唯一可行的方法就是使用并行或 ICP 编程，假设用户应用程序不包含可擦除和编程引导向量和引导状态位的自定义装载程序。在对 Flash 进行编程后，状态字节应编程为 0，以允许用户应用代码从地址 0000H 开始执行。

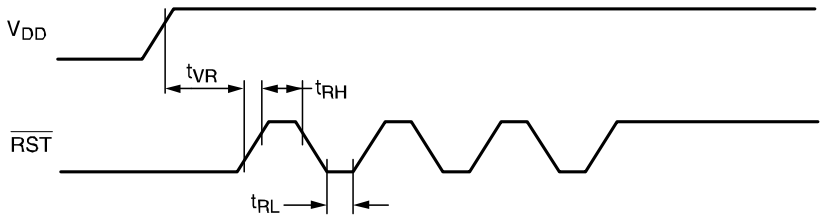


图 45　强制进入 ISP 模式

17.10 在系统编程(ISP)

执行 ISP 编程时不需要将微控制器从系统中移出。在系统编程特性包含了一系列内部的硬件资源，与内部固件相结合可实现通过串口对 P89LPC952 的远程编程。固件由 PHILIPS 提供并嵌入到每一个 P89LPC952 芯片当中。PHILIPS 的在系统编程特性使嵌入式应用中的在电路编程变得可行，并最大限度减少了额外的元件开销和电路板面积。ISP 功能使用芯片的 5 个管脚(V_{DD}、Vss、TxD0、RxD0、$\overline{RST}$)。使用该特性时只需要一个小的连接器将您的应用与外部电路连接。

17.11 使用在系统编程(ISP)

ISP 特性可以适应用户应用中较宽范围的波特率而不依赖于振荡器的频率。它还可以适应较宽范围的振荡器频率。这是通过对一个接收到的字符中单个位进行位时间测量来实现的。然后，根据基于振荡器频率的定时器值该信息用于对波特率进行编程。为此 ISP 特性要求主机必须向 P89LPC952 发送一个初始化字符（大写的 U）以设定芯片的波特率。ISP 固件提供对接收字节的自动应答。一旦波特率的设定完成之后，ISP 固件将只接受 Intel hex 格式的记录。它包含了用于表示十六进制数的 ASCII 字符。汇总如下：

：NNAAAARRDD..DDCC<crlf>

在 Intel Hex 格式中，‘NN’表示记录中数据字节的个数。P89LPC952 最多可接受 64（40H）个数据字节。‘AAAA’字符串表示记录中首字节的地址。如果记录中的字节数为 0，通常该区域设置为‘0000’。‘RR’表示记录的类型：‘00’表示数据类型，‘01’表示文件结束标识。在该应用中，将增加额外的记录格式以指示 ISP 应用的命令或数据。数据字节的最大数目限制为 64（十进制）。ISP 命令汇总见表 98。P89LPC952 收到记录后，将数据保存并计算校验和。在整个记录接收完毕后才执行由记录类型所指示的操作。如果校验和发生错误，P89LPC952 将从串口发出‘X’指示校验和出错。如果校验结果与记录相符，将会执行指定的命令操作。在大多数情况下，从串口发出‘.’字符表示正确接收。

表 98　在系统编程（ISP）Hex 记录格式

记录类型	命令/数据 功能
00	编程用户代码存储器页 :nnaaaa00dd....ddcc 其中: nn　　= 编程的字节数 aaaa　　= 页地址 dd....dd　= 数据字节 cc　　= 校验和 例：:10000000010203040506070809cc
01	读版本 Id :00xxxx01cc 其中: xxxx　　= 必须的字段，但其值“无关” cc　　= 校验和 例：:00000001cc
02	混合写功能 :02xxxx02ssddcc 其中: xxxx　　= 必须的字段，但其值“无关” ss　　= 子功能代码 dd　　= 数据 cc　　= 校验和 子功能代码: 00　　= UCFG1 01　　= 保留 02　　= 引导向量 03　　= 状态字节 04　　= 保留 05　　= 保留 06　　= 保留 07　　= 保留 08　　= 保密字节 0 09　　= 保密字节 1 0A　　= 保密字节 2 0B　　= 保密字节 3 0C　　= 保密字节 4 0D　　= 保密字节 5 0E　　= 保密字节 6 0F　　= 保密字节 7 0A　　= 清除配置保护 例：:020000020347cc

续表 98

记录类型	命令/数据 功能
03	混合读功能 :01xxxx03sscc 其中: xxxx = 必须的字段，但其值“无关” ss = 子功能代码 cc = 校验和 子功能代码: 00 = UCFG1 01 = 保留 02 = 引导向量 03 = 状态字节 04 = 保留 05 = 保留 06 = 保留 07 = 保留 08 = 保密字节 0 09 = 保密字节 1 0A = 保密字节 2 0B = 保密字节 3 0C = 保密字节 4 0D = 保密字节 5 0E = 保密字节 6 0F = 保密字节 7 10 = 厂商 ID 11 = 器件 ID 12 = 派生器件 ID 例: ：0100000312cc
04	擦除扇区/页 ：03xxxx04ssaaaacc 其中 xxxx = 必须的字段，但其值“无关” aaaa = 扇区/页地址 ss = 01 擦除扇区 = 00 擦除页 cc = 校验和 例: ：03000004010000F8

续表 98

记录类型	命令/数据 功能
05	读扇区 CRC ：01xxxx05aacc 其中： xxxx = 必须的字段，但其值“无关” aa = 扇区地址高字节 cc = 校验和 例： :0100000504F6cc
06	读全局 CRC ：00xxxx06cc 其中： xxxx = 必须的字段，但其值“无关” cc = 校验和 例 :00000006FA
07	直接装载波特率 ：02xxxx07HHLLcc 其中: xxxx = 必须的字段，但其值“无关” HH = 定时器高字节 LL = 定时器低字节 cc = 校验和 例： :02000007FFFFcc
08	复位 MCU ：00xxxx08cc 其中: xxxx = 必须的字段，但其值“无关” cc = 校验和 例: :00000008F8

17.12 在应用中编程的方法(IAP)

通过在应用程序中对几个 IAP 子程序的调用，可实现有选择地对 Flash 扇区、页、保密位、配置字节和器件 ID 进行擦除和编程。所有的调用都是通过一个公共的接口 PGM_MTP 来实现的。在调用位于地址 FF03H 的 PGM_MTP 之前，通过设置微控制器的寄存器可选择编程的功能。IAP 调用见表 100。

17.13 IAP 特许关键值

执行写或擦除代码存储器的 IAP 功能需要设置一个特许关键值，该关键值在执行 IAP 功能调用之前由调用程序来设置。通过将 96H 写入 RAM 的 FFH 单元完成关键值的设置。下面的例程使用的是 KeilC 编译器。访问存储器中指定物理地址的方法会随着编译器的不同而不同。

```
Include <ABSACC.H>                    / *使能绝对内存地址的访问*/
Define key DBYTE [0xFF]               /*强制关键值的地址为 0xFF*/
Short (*pgm_mtp) (void) =0xFF00;    / *设置指针指向 IAP 入口处*/
Key=0x96;                            /*  设置特许关键值*/
pgm_mtp ( );                         / *执行 IAP 功能调用*/
```

IAP 程序处理完功能调用后，特许关键值被清除。因此，在 EACH 调用需要关键值的 PGM_MTP 前必须设置特许关键值。如果需要特许关键值的 IAP 程序被调用但又无有效特许关键值存在时，MCU 将复位。

17.14 Flash 写使能

LPC952 含有硬件写使能保护功能。该功能可应用于 ISP 和 IAP 模式以及用户程序存储器空间和用户配置字节（UCFG1,BOOTVEC 和 BOOTSTAT），但不能用于 ICP 或并行编程模式。如果 BOOTSTAT.7 的激活写使能位（AWE）位为 0，则内部写使能标志（WE）被强制置位，使能 Flash 存储器和配置字节的写操作。如果激活写使能（AWE）位为 1，内部 WE 标志的状态由用户控制。

WE 标志通过先向 FMCON 写入设置写使能（08H）命令，再将关键值（96H）写入 FMDATA 来置位：

FMCON＝0x08

FMDATA＝0x96

WE 标志通过先向 FMCON 写入清零写使能（0BH）命令，再将关键值（96H）写入 FMDATA 或复位来清零：

FMCON＝0x0B

FMDATA＝0x96

ISP 功能在调用 IAP 程序之前置位 WE 标志。IAP 功能在任意写操作完成后执行一次清除写使能命令。如果写使能功能激活，要调用 IAP 程序的用户代码每次执行 IAP 写功能调用之前都必须将写使能标志置位。

17.15 配置字节保护

除上述的硬件写使能保护外，器件还可对“配置字节”进行单独写保护。这些配置字节包括 UCFG1, BOOTVEC 和 BOOTSTAT。该保护功能应用于 ISP 和 IAP 模式，但不能用于 ICP 或并行编程模式。

如果 BOOTSTAT.6 的配置写保护位（CWP）为 1，则禁止配置字节的写操作。如果 BOOTSTAT.6 的配置写保护位（CWP）为 0，则使能配置字节的写操作。CWP 位通过编程 BOOTSTAT 寄存器来置位，通过 IAP 或 ISP 中的清除配置保护功能来清零。

ISP 或 IAP 模式中的清除配置保护命令通过向 BOOTSTAT.7 的禁止清零配置保护位（DCCP）写入 1 来禁能。DCCP 位置位时，ICP 或并行编程模式可继续使用 CCP 命令。DCCP 位通过在 ICP 或并行编程模式中写入清除配置保护命令来清零。

17.16 IAP 错误状态

程序指令是不可能存放在正在编程或擦除的 Flash 存储器中的。因此，在 IAP 擦除、编程或 CRC 周期内，CPU 进入编程-空闲状态。CPU 将一直处于该状态直至擦除、编程或 CRC 周期结束。这些周期是自定时的。当擦除、编程或 CRC 周期结束后，恢复代码的执行。如果在执行擦除、编程或 CRC 操作时出现中断，这些操作被终止，Flash 存储器用作代码存储器去响应中断。置位进位标志并返回状态信息，表示 IAP 出错。返回的状态信息如表 99 所示。如果允许在擦除、编程或 CRC 周期内发生中断，用户程序必须在每次操作后检查进位标志来判断是否发生错误。如果操作被终止，用户代码必须使操作重复执行。

表 99 IAP 错误状态

位	标志	描述
0	OI	操作被中断。由于编程或擦除周期内发生中断而使操作终止。
1	SV	安全出错。该位在由于安全设置而引起编程或擦除操作失败时置位。操作终止，存储器内容不变，CRC 输出无效。
2	HVE	高电压错误。高电压发生电路出错时该位置位。操作终止，存储器内容可能被破坏。
3	VE	校验出错。当用户代码执行 IAP 编程过程时，被编程地址的内容和预定编程值不相同时该位置位。IAP 利用 MOVC 指令进行校验。允许对 MOVC 被保护的用户程序进行编程，但编程结束后将产生错误标志。
4-7	-	未用；读出为‘0’。

表 100　IAP 功能调用

IAP 功能	IAP 调用参数
编程用户代码页 （需要‘关键值’）	输入参数： ACC = 00h R3　= 编程字节数 R4　= 页地址（高字节） R5　= 页地址（低字节） R7　= RAM 中数据缓冲区的指针 F1　= 0h =使用 IDATA 返回参数 R7　= 状态 进位位（C）= 置位表示错误，清零表示无错误
读版本 ID	输入参数：ACC = 01h 返回参数：R7　= IAP 代码版本 ID
混合写 （需要‘关键值’）	输入参数： ACC = 02h R5　= 写入的数据 R7　= 寄存器地址 00　= UCFG1 01　= 保留 02　= 引导向量 03　= 状态字节 04　= 保留 05　= 保留 06　= 保留 07　= 保留 08　= 保密字节 0 09　= 保密字节 1 0A　= 保密字节 2 0B　= 保密字节 3 0C　= 保密字节 4 0D　= 保密字节 5 0E　= 保密字节 6 0F　= 保密字节 7 10　=清除配置保护 返回参数 R7　= 状态 进位位（C）= 置位表示错误，清零表示无错误

续表 100

IAP 功能	IAP 调用参数
混合读	输入参数： ACC = 03h R7 = 寄存器地址 00 = UCFG1 01 = 保留 02 = 引导向量 03 = 状态字节 04 = 保留 05 = 保留 06 = 保留 07 = 保留 08 = 保密字节 0 09 = 保密字节 1 0A = 保密字节 2 0B = 保密字节 3 0C = 保密字节 4 0D = 保密字节 5 0E = 保密字节 6 0F = 保密字节 7 返回参数 R7 = 如果无错误则为寄存器数据，否则为错误状态 进位位（C）= 置位表示错误，清零表示无错误
擦除扇区/页 （需要‘关键值’）	输入参数： ACC = 04h R7 = 00H(擦除页)或 01H(擦除扇区) R4 = 扇区/页地址（高字节） R5 = 扇区/页地址（低字节） 返回参数 R7 = 状态 进位位（C）= 置位表示错误，清零表示无错误
读扇区 CRC	输入参数： ACC = 05h R7 = 扇区地址 返回参数： R4 = CRC 位 31:24 R5 = CRC 位 23:16 R6 = CRC 位 15:8 R7 = CRC 位 7:0(如果无错误) R7 = 错误状态（如果错误） 进位位（C）= 置位表示错误，清零表示无错误

续表 100

IAP 功能	IAP 调用参数
读全局 CRC	输入参数： ACC = 06h 返回参数： R4 = CRC 位 31:24 R5 = CRC 位 23:16 R6 = CRC 位 15:8 R7 = CRC 位 7:0(如果无错误) R7 = 错误状态（如果错误） 进位位（C）= 置位表示错误，清零表示无错误
读用户代码	输入参数： ACC = 07h R4 = 地址(高字节) R5 = 地址(低字节) 返回参数 R7 = 数据

17.17 用户配置字节

P89LPC952 的众多用户可配置的特性必须在上电时定义，开始执行程序后便不可设置了。这些特性是通过配置 Flash 字节 UCFG1 实现，如表 102 所示。

表 101 Flash 用户配置字节（UCFG1）位分配

位	7	6	5	4	3	2	1	0
符号	WDTE	RPE	BOE	WDSE	-	FOSC2	FOSC1	FOSC0
未编程值	0	1	1	0	0	0	1	1

表 102 Flash 用户配置字节（UCFG1）位描述

位	符号	描述
0	FOSC0	CPU 振荡器类型选择。详见第 2 节“时钟”。除了表 105 给出的组合外，其它值不能使用。它们被保留供将来使用。
1	FOSC1	
2	FOSC2	
3	-	保留
4	WDSE	看门狗安全使能位。详见表 189“看门狗定时器配置”。
5	BOE	掉电检测使能（见 6.1 节“掉电检测”）。
6	RPE	复位脚使能，当置 1 时，使能 P1.5 脚的复位功能。清零时，P1.5 可作为输入口使用。注：在上电过程中，RPE 选项无效，该管脚总是作为复位输入管脚。在上电之后，该管脚执行 RPE 位所定义的功能。只有上电复位才能暂时使 RPE 位的设定无效。其它复位源无法影响 RPE 位的设定。
7	WDTE	看门狗定时器复位使能位，置位时使能看门狗定时器的复位，清零时禁止看门狗定时器的复位。定时器仍可用于产生中断。详见表 89“看门狗定时器配置”。

表 103 Flash 用户配置字节 2（UCFG2）位分配

位	7	6	5	4	3	2	1	0
符号	DBG_A	-	DBG_B	HLT_T1	HLT_RTC	HLT_T0	-	-
未编程值	x	x	0	0	0	0	x	x

表 104 Flash 用户配置字节 2（UCFG2）位描述

位	符号	描述
0:1	-	未使用
2	HLT_T0	中止定时器 0。对该位编程后，当器件进入调试状态时定时器 0 中止。
3	HLT_RTC	中止定时器 RTC。对该位编程后，当器件进入调试状态时 RTC 中止。
4	HLT_T1	中止定时器 1。对该位编程后，当器件进入调试状态时定时器 1 中止。
5	DBG_B	调试使能 B。当对 DBG_A 和 DBG_B 进行编程后，使能器件进入调试模式。
6	TRIGEN	触发输出使能。对该位编程后，使能 TRIG 管脚的触发输出。
7	DBG_A	调试输出 B。当对 DBG_A 和 DBG_B 进行编程后，使能器件进入调试模式。

表 105　振荡器类型选择

FOSC[2:0]	振荡器配置
111	从 XTAL1 输入的外部时钟
100	看门狗振荡器，400KHz（+20%, -30%误差）
011	内部 RC 振荡器，7.373MHz±2.5%
010	低频晶振，20KHz 至 100KHz
001	中频晶振或谐振器，100KHz 至 4MHz
000	高频晶振或谐振器，4MHz 至 18MHz

17.18 用户保密字节

P89LPC952 有 3 个保密位，与 8 个扇区对应，如表 106 所示：

表 106　扇区保密字节（SECx）位分配

位	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	EDISx	SPEDISx	MOVCDISx
未编程值	0	0	0	0	0	0	0	0

表 107　扇区保密字节（SECx）位描述

位	符号	描述
0	MOVCDISx	MOVC 禁止。禁止对扇区 x 执行 MOVC 命令。任何对 MOVC 指令受保护的扇区进行 MOVC 访问都将返回无效数据。只有当扇区 x 被擦除时才能将该位擦除。
1	SPEDISx	扇区编程擦除禁止 x。禁止对扇区 x 的全部或部分字节进行编程或擦除。该位和扇区 x 可通过扇区擦除命令（ISP,IAP,商用编程器）或‘整体’擦除命令（商用编程器）进行擦除。

续表 107

位	符号	描述
2	EDISx	禁止 ISP 擦除。禁止在 ISP 或 IAP 模式下对扇区 x 进行擦除。该位经编程后，它和扇区 x 只可能使用商用编程器执行‘整体’擦除命令进行擦除。在 ISP 或 IAP 模式中，不能将该位和扇区 x 进行擦除。
3:7	-	保留

表 108　加密位的作用

EDISx	SPEDISx	MOVCDISx	编程时的影响
0	0	0	无。
0	0	1	对指定扇区进行扇区 CRC 计算时安全出错标志置位；如果 MOVCDISx 位置位，整体 CRC 校验的安全出错标志置位；操作周期终止；存储器内容不变；CRC 无效；执行编程/擦除命令不会造成安全出错。
0	1	X	执行编程命令或擦除页命令时安全出错标志置位；操作周期终止；存储器内容不变；允许扇区擦除和整体擦除。
1	X	X	执行编程或擦除页命令时置位安全出错标志；操作周期终止；存储器内容不变；允许整体擦除。

17.19 引导向量寄存器

表 109　引导向量（BOOTVEC）位分配

位	7	6	5	4	3	2	1	0
符号	-	-	-	BOOTV4	BOOTV3	BOOTV2	BOOTV1	BOOTV0
出厂默认值	0	0	0	1	1	1	1	1

表 110　引导向量（BOOTVEC）位描述

位	符号	描述
0:4	BOOTV[0:4]	引导向量。如果选择引导向量作为复位地址，P89LPC952 复位后将从 BOOTVEC 作为地址高字节，00h 作为地址低字节形成的地址处开始执行程序。
5:7	-	保留

17.20 引导状态寄存器

表 111　引导状态（BOOTSTAT）位分配

位	7	6	5	4	3	2	1	0
符号	DCCP	CWP	AWP	-	-	-	-	BSB
出厂默认值	0	0	0	0	0	0	0	1

表 112　引导状态（BOOTSTAT）位描述

位	符号	描述
0	BSB	引导状态位。如果该位编程为 1，P89LPC952 复位后将始终从 BOOTVEC 作为地址高字节，00h 作为地址低字节形成的地址处开始执行程序（见 7.1 节“复位向量”）。
1:4	-	保留
5	AWP	激活写保护位。该位清零时，内部写使能标志强制置位，这时，写 Flash 存储器始终使能。如果该位置位，内部写使能标志通过使用设置写使能（SWE）命令和清除写使能（CWE）命令来置位和清零。
6	CWP	配置写保护位。保护用户可编程配置字节（UCFG1,BOOTVEC 和 BOOTSTAT），防止产生无意的写操作。如果该位为 1，这些配置字节寄存器的写操作被禁止。如果该位为 0，这些寄存器的写操作被使能。 该位通过编程 BOOTSTAT 寄存器来置位，通过先向 FMCON 写入清除配置保护（CCP）命令再将 96H 写入 FMDATA 来清零。
7	DCCP	禁能清除配置保护命令。如果该位为 1，清除配置保护（CCP）命令在 ISP 或 IAP 模式下被禁止，但它仍可用于 ICP 或并行编程模式。如果该位为 0，CCP 命令在所有模式下都可使用。该位通过编程 BOOTSTAT 寄存器置位，通过在 ICP 或并行编程模式中写入清除配置保护命令来清零。

18.调试接口

P89LPC952 包含一个调试（JTAG）接口，它能够与商用调试器连接。器件进入调试模式之前，必须具备以下条件：

- UCFG2.5 和 UCFG2.7 已经编程
- SCLK 与一个连接到 V_{DD} 的 10KΩ 上拉电阻相连
- SDAT 与一个连接到 V_{DD} 的 10KΩ 上拉电阻相连
- 调试管脚与商用编程器相连
- 发生上电复位或外部复位

在调试模式中，定时器 0/1 和 RTC 可以中止。见表 104“Flash 用户配置字节 2（UCFG2）位描述”。

在进行调试时，还需要一个工业标准的 10 管脚，间距为 0.100 英寸的双槽公座连接头作为调试连接器。

表 113 调试（JTAG）接口的连接器管脚分配

管脚	功能	管脚	功能
2	TRIG（触发输出）	1	V_{DD}
4	SDAT（串行数据）	3	V_{SS}
6	$\overline{\textbf{RST}}$（MCU 复位）	5	
8	SCLK（串行时钟）	7	
10	未用	9	

19.指令集

表 114　指令集

助记符	描述	字节	周期	Hex 码
算术指令				
ADD A,Rn	寄存器的内容加到 A	1	1	28-2F
ADD A,dir	直接寻址字节的内容加到 A	2	1	25
ADD A,@Ri	间接寻址存储器单元的内容加到 A	1	1	26-27
ADD A,#data	立即数加到 A	2	1	24
ADDC A,Rn	寄存器的内容带进位位加到 A	1	1	38-3F
ADDC A,dir	直接寻址单元的内容带进位位加到 A	2	1	35
ADDC A,@Ri	间接寻址存储单元的内容带进位位加到 A	1	1	36-37
ADDC A,#data	立即数带进位位加到 A	2	1	34
SUBB A,Rn	从寄存器 A 中减去寄存器的内容，可借位	1	1	98-9F
SUBB A,dir	从寄存器 A 中减去直接寻址单元的内容，可借位	2	1	95
SUBB A,@Ri	从寄存器 A 中减去间接寻址存储单元的内容，可借位	1	1	96-97
SUBB A,#data	从寄存器 A 中减去立即数，可借位	2	1	94
INC A	寄存器 A 的值加 1	1	1	04
INC Rn	寄存器的值加 1	1	1	08-0F
INC dir	直接寻址单元的内容加 1	2	1	05
INC @Ri	间接寻址存储单元的内容加 1	1	1	06-07
DEC A	寄存器 A 的值减 1	1	1	14
DEC Rn	寄存器的值减 1	1	1	18-1F
DEC dir	直接寻址单元的内容减 1	2	1	15
DEC @Ri	间接寻址存储单元的内容减 1	1	1	16-17
INC DPTR	数据指针加 1	1	2	A3
MUL AB	A 和 B 相乘，乘积放在 A 中	1	4	A4
DIV AB	A 除以 B，商放在 A 中	1	4	84
DA A	A 的内容十进制调整	1	1	D4
逻辑指令				
ANL A,Rn	寄存器与 A 相与，结果放在 A 中	1	1	58-5F
ANL A,dir	直接寻址单元与 A 相与，结果放在 A 中	2	1	55
ANL A,@Ri	间接寻址存储单元与 A 相与，结果放在 A 中	1	1	56-57
ANL A,#data	立即数与 A 相与，结果放在 A 中	2	1	54
ANL dir,A	直接寻址单元与 A 相与，结果放在直接寻址单元中	2	1	52
ANL dir,#data	立即数与直接寻址单元相与，结果放在直接寻址单元中	3	2	53
ORL A,Rn	寄存器与 A 相或，结果放在 A 中	1	1	48-4F
ORL A,dir	直接寻址单元与 A 相或，结果放在 A 中	2	1	45
ORL A,@Ri	间接寻址存储单元与 A 相或，结果放在 A 中	1	1	46-47
ORL A,#data	立即数与 A 相或，结果放在 A 中	2	1	44
ORL dir,A	直接寻址单元与 A 相或，结果放在直接寻址单元中	2	1	42
ORL dir,#data	立即数与直接寻址单元相或，结果放在直接寻址单元中	3	2	43

续表 114

助记符	描述	字节	周期	Hex 码
XRL A,Rn	寄存器与 A 相异或，结果放在 A 中	1	1	68-6F
XRL A,dir	直接寻址单元与 A 相异或，结果放在 A 中	2	1	65
逻辑指令				
XRL A,@Ri	间接寻址存储单元与 A 相异或，结果放在 A 中	1	1	66-67
XRL A,#data	立即数与 A 相异或，结果放在 A 中	2	1	64
XRL dir,A	直接寻址单元与 A 相异或，结果放在直接寻址单元中	2	1	62
XRL dir,#data	立即数与直接寻址单元相异或，结果放在直接寻址单元中	3	2	63
CLR A	A 清零	1	1	E4
CPL A	A 取反	1	1	F4
SWAP A	A 的内容高低半字节交换	1	1	C4
RL A	A 的内容循环左移	1	1	23
RLC A	A 的内容带进位位循环左移	1	1	33
RR A	A 的内容循环右移	1	1	03
RRC A	A 的内容带进位位循环右移	1	1	13
数据传送				
MOV A,Rn	寄存器的数据传送给 A	1	1	E8-EF
MOV A,dir	直接寻址单元的内容传送给 A	2	1	E5
MOV A,@Ri	间接寻址存储单元的内容传送给 A	1	1	E6-E7
MOV A,#data	立即数传送给 A	2	1	74
MOV Rn,A	A 的内容传送给寄存器	1	1	F8-FF
MOV Rn,dir	直接寻址单元的内容传送给直接寻址字节	2	2	A8-AF
MOV Rn,#data	立即数传送给寄存器	2	1	78-7F
MOV dir,A	A 的内容传送给直接寻址单元	2	1	F5
MOV dir,Rn	寄存器的内容传送给直接寻址单元	2	2	88-8F
MOV dir,dir	直接寻址单元的内容传送给直接寻址单元	3	2	85
MOV dir,@Ri	间接存储单元的内容传送给直接寻址单元	2	2	86-87
MOV dir,#data	立即数传送给直接寻址单元	3	2	75
MOV @Ri,A	A 的内容传送给间接寻址存储单元	1	1	F6-F7
MOV @Ri,dir	直接寻址单元的内容传送给间接寻址存储单元	2	2	A6-A7
MOV @Ri,#data	立即数传送给间接寻址存储单元	2	1	76-77
MOV DPTR,#data	立即数传动给数据指针	3	2	90
MOVC A,@A+DPTR	DPTR 相对地址单元的代码字节传送给 A	1	2	93
MOVC A,@A+PC	PC 相对地址单元的代码字节传送给 A	1	2	94
MOVX A,@Ri	外部数据（A8）传送给 A	1	2	E2-E3
MOVX A,@DPTR	外部数据（A16）传送给 A	1	2	E0
MOVX @Ri,A	A 的内容传送到外部数据单元（A8）	1	2	F2-F3
MOVX @DPTR,A	A 的内容传送到外部数据单元（A16）	1	2	F0
PUSH dir	直接寻址单元的内容压栈	2	2	C0
POP dir	直接寻址单元的内容出栈	2	2	D0
XCH A,Rn	交换 A 和寄存器的内容	1	1	C8-CF

续表 114

助记符	描述	字节	周期	Hex 码
XCH A,dir	交换 A 和直接寻址单元的内容	2	1	C5
XCH A,@Ri	交换 A 和间接寻址存储单元的内容	1	1	C6-C7
XCHD A,@Ri	交换 A 和间接寻址存储单元半字节的内容	1	1	D6-D7
位处理指令				
CLR C	进位位清零	1	1	C3
CLR bit	直接寻址位清零	2	1	C2
SETB C	进位位置位	1	1	D3
SETB bit	直接寻址位置位	2	1	D2
CPL C	进位位取反	1	1	B3
CPL bit	直接寻址位取反	2	1	B2
ANL C,bit	直接寻址位与进位位相与，结果放在进位位中	2	2	82
ANL C,/bit	直接寻址位取反后与进位位相与，结果放在进位位中	2	2	B0
ORL C,bit	直接寻址位与进位位相或，结果放在进位位中	2	2	72
ORL C,/bit	直接寻址位取反后与进位位相或，结果放在进位位中	2	2	A0
MOV C,bit	直接寻址位的值传送给进位位	2	1	A2
MOV bit,C	进位位传送给直接寻址位	2	2	92
转移指令				
ACALL addr11	子程序的绝对调用	2	2	116F1
LCALL addr16	长调用到子程序	3	2	12
RET	子程序返回	1	2	22
RETI	中断返回	1	2	32
AJMP addr11	无条件绝对跳转	2	2	016E1
LJMP addr16	无条件长跳转	3	2	02
SJMP rel	短跳转（相对地址）	2	2	80
JC rel	进位位＝1 时跳转	2	2	40
JNC rel	进位位＝0 时跳转	2	2	50
JB bit,rel	直接寻址位＝1 时跳转	3	2	20
JNB bit,rel	直接寻址位＝0 时跳转	3	2	30
JBC bit,rel	直接寻址位＝1 时跳转，然后清零	3	2	10
JMP @A+DPTR	跳转到 DPTR 相对地址指向的地址	1	2	73
JZ rel	累加器＝0 时跳转	2	2	60
JNZ rel	累加器≠0 时跳转	2	2	70
CJNE A,dir,rel	比较 A 和直接寻址单元的内容,不等时转移到相对地址	3	2	B5
CJNE A,#d,rel	比较 A 和立即数，不等时转移到相对地址	3	2	B4
CJNE Rn,#d,rel	比较寄存器和立即数，不等时转移到相对地址	3	2	B8-BF
CJNE @Ri,#d,rel	比较间接寻址单元和立即数，不等时转移到相对地址	3	2	B6-B7
DJNZ Rn,rel	寄存器减 1，不为零时转移到相对地址	2	2	D8-DF
DJNZ dir,rel	直接寻址单元减 1，不为零时转移到相对地址	3	2	D5
其它指令				
NOP	空操作	1	1	00

20.修订记录

表 115　修订记录

版本	日期	描述
01	2005 年 9 月 12 日	最初版本